

Tietojärjestelmien kehittäminen

Case Kelan JAMES-systeemityömenetelmä

Miia Nopanen



Tekijä Miia Nopanen	Ryhmätunnus tai aloitusvuosi TMI08
Raportin nimi Tietojärjestelmien kehittäminen Case Kelan JAMES-systeemityömenetelmä	Sivu- ja liitesivumäärä 48 + 87
Opettajat tai ohjaajat Sirpa Marttila	
<p>Tämä toiminnallinen opinnäytetyö tehtiin toimeksiantona Kansaneläkelaitoksen IT-osastolle vuosien 2011 ja 2012 aikana. Työn aiheena on tietojärjestelmien kehittäminen, ja työn teoriaosuuden tavoitteena on esitellä systeemityöhön liittyvät vaiheet esitutkimusvaiheesta ylläpitovaiheeseen. Näiden vaiheiden keskinäinen järjestys ja ajoitus määritellään vaihejakomallissa. Teoriaosuudessa esitellään myös kolme erilaista vaihejakomallia: vesiputousmalli, prototyypilähestymistapa ja spiraalimalli.</p> <p>Työn toimeksiantajan tehtäväkenttä painottuu erilaisten tilastoinnin ja raportoinnin järjestelmien ylläpitoon ja kehittämiseen SAS-teknologian avulla. Näiden järjestelmien kehitystyössä on aiemmin sovellettu löyhästi vesiputousmallia, jota ei kuitenkaan ole selkeästi standardoitu ohjaamaan sovelluskehittäjien toimintaa. Toimeksiantajaorganisaatiossa on kehitetty oma hyvin standardoitu systeemityömenetelmä JAMES, joka on alun perin suunnattu lähinnä JAVA-teknologialla toteutettavaan sovelluskehitystyöhön. Toimeksiantaja tarvitsi tietoa siitä, kuinka JAMES-menetelmää voisi hyödyntää myös SAS-teknologialla toteuttavien datankäsittelyyn liittyvien tilastoinnin ja raportoinnin järjestelmien kehittämisessä.</p> <p>Opinnäytetyön toiminnallisen osion tavoitteena oli suorittaa uuden tilastointiin ja raportointiin liittyvän tietojärjestelmän kehitystyön aloitusvaihe JAMES-systeemityömenetelmää soveltaen ja tuottaa menetelmän mukaista dokumentaatiota. Opinnäytetyöprosessin aikana tuotettiin kuusi erilaista määrittelydokumenttia. Nämä dokumentit toimivat sekä kehitettävän järjestelmän määrittelydokumentaationa että mallipohjina tuleville vastaavanlaisille tilastointi- ja raportointijärjestelmien kehittämisprojekteille. Tuotetut dokumentit on sijoitettu opinnäytetyön liitteisiin.</p>	
Asiasanat systeemityö, tietojärjestelmät, dokumentointi	

Degree programme in Business Information Technology

Authors Miia Nopanen	Group or year of entry TMI08
The title of thesis The Information Systems Development Case JAMES Systems Development Method in Kela	Number of pages and appendices 48+ 87
Supervisor(s) Sirpa Marttila	
<p>This functional thesis was assigned by the IT department of the Social Insurance Institution of Finland, Kansaneläkelaitos. The topic of the thesis is the development of information systems and it was conducted during the years 2011 and 2012. The objective of the theoretical section is to explain the phases that systems development contains from the feasibility study phase to the maintenance phase. The order and timing of these phases is specified in systems development life cycle. The theoretical section presents three different life cycles: the waterfall model, the prototype model and the spiral model.</p> <p>The main task of this thesis' client is to maintain and to develop different statistical analyses and reporting systems using mainly SAS technology. The development of these systems has previously been loosely applied to the waterfall model. The problem is that the model is not well standardized in the organization to control and to guide the operation of the application developers. The organization has developed its own well-standardized systems development method JAMES. It was originally directed primarily at application development carried out by JAVA technology, but it could be useful within client's other technologies too. The client needed information about how the JAMES-method could be utilized for the development of the statistical analyses and reporting systems within SAS technology.</p> <p>The objective of the functional section of this thesis was to perform the initial phase of a new statistical analysis and reporting system developed with SAS technology utilizing the JAMES method. The work was emphasized in producing certain specification documentation according to the JAMES method. The actual product delivered to the client was six different documents concerning the new information system. These documents serve as documentation for the system to be used in the further development as well as for an example for similar projects in the future. The produced documents are attached in the appendix of this thesis.</p>	
Key words systems development, information systems, documentation	

Sisällys

1	Johdanto	1
1.1	Opinnäytetyön tavoitteet ja rajaukset	2
1.2	Opinnäytetyön rakenne	3
2	Tietojärjestelmien kehittäminen	4
2.1	Systeemityön vaihejako.....	4
2.2	Systeemityön vaihejakomallit	10
3	Toimeksiantajaorganisaatio	16
3.1	Organisaation esittely.....	16
3.2	Toimintaympäristön esittely.....	17
4	Toimeksiantajan JAMES-systeemityömenetelmä	19
4.1	JAMES-menetelmän rakenneosat	20
4.2	JAMES-menetelmään liittyvät tehtävät	24
5	Toimeksiannon toteuttaminen	26
5.1	Toimeksiannon esittely	27
5.2	Kehitysprojektin Aloitusvaiheen toteuttaminen ja dokumentointi	28
6	Johtopäätökset ja pohdinta	37
6.1	Yhteenvedo	37
6.2	Opinnäytetyöprosessi.....	44
	Lähteet.....	47
	Liitteet.....	49
	Liite 1. Opinnäytetyössä käytetyt termit ja käsitteet	49
	Liite 2. JAMES-lopputuote Sidosryhmän tarpeet.....	56
	Liite 3. JAMES-lopputuote Visio.....	76
	Liite 4. JAMES-lopputuote Sanasto	106
	Liite 5. JAMES-lopputuote Järjestelmän yleiskuvaus	114
	Liite 6. JAMES-lopputuote Tietoturvaselvitys	121
	Liite 7. JAMES-lopputuote Systeemitestisuunnitelma	129

1 Johdanto

Yrityksissä ja muissa organisaatioissa käsitellään suurta määrää erilaista tietoa. Työtä helpottamaan on rakennettu erilaisia tietojärjestelmiä, jotka suorittavat niille määritellyjä tehtäviä, kuten tilausten hallintaa tai rekisterien ylläpitoa. Organisaation tarpeet ja ympäristön odotukset kuitenkin muuttuvat ajan myötä, jolloin tietojärjestelmä ei ehkä enää suoriudukaan tehtävistään toivotulla tavalla. Organisaatiossa joudutaan päättämään, uudistetaanko vanhaa tietojärjestelmää vai kehitetäänkö sen tilalle kokonaan uusi tietojärjestelmä. Päädyttiin sitten kumpaan vaihtoehtoon tahansa, kehitystyön tulee olla suunnitelmallista, järjestelmällistä ja hyvin dokumentoitua. Tällaista systemaattista tietojärjestelmien kehitystyötä kutsutaan systeemityöksi.

Systeemityön periaatteiden mukaisesti tietojärjestelmien kehitystyötä suoritetaan vaiheittain ennalta valitun mallin mukaisesti. Yleisiä käytössä olevia malleja ovat esimerkiksi vesiputousmalli, spiraalimalli ja RUP. Tässä toiminnallisessa opinnäytetyössäni sovelsin toimeksiantajaorganisaatiossa kehitettyä, RUP:iin pohjautuvaa JAMES-systeemityömenetelmää uuden SAS-teknologialla toteutettavan tilastointi- ja raportointijärjestelmän kehitystyössä. Opinnäytetyöprosessin aikana suoritin kehitystyön ensimmäisen eli Aloitusvaiheen ja tuotin kuusi JAMES-menetelmän mukaista määrittelydokumenttia. Tulen jatkamaan aloittamaani kehitystyötä vielä opinnäytetyön jälkeen. Tarkoitus on, että vuoden 2013 alussa toimeksiantajalla olisi toimivan tietojärjestelmän lisäksi JAMES-menetelmän mukaiset mallipohjat, joita voitaisiin jatkossa hyödyntää vastaavanlaisissa kehittämisprojekteissa.

Toimeksiantajalle luovutetuissa dokumenteissa määritelty tietojärjestelmä liittyy toimeksiantajaorganisaation tietoarkkitehtuurin kehittämiseen. Organisaatiossa käsiteltävien tietojen tietokuvauksia eli metadataa on koottu tietokantaan. Tavoitteena on, että kantaan tallennettuja kuvaustekstejä voitaisiin tulevaisuudessa hyödyntää ohjelmallisesti toimeksiantajan tilastointi- ja raportointijärjestelmissä. Tämä ei kuitenkaan toistaiseksi ole mahdollista, sillä soveltuvaa menetelmää ei ole. Tarvittaisiin tietojärjestelmä, jonka avulla tietokantaan tallennetut kuvaustekstit saataisiin muiden järjestelmien käyttöön. Tämän tietojärjestelmän kehittämistarve oli pohjana opinnäytetyölleni.

1.1 Opinnäytetyön tavoitteet ja rajaukset

Tavoitteenani opinnäytetyön teoreettisessa osuudessa oli selvittää, mitkä ovat tietojärjestelmäkehitysprosessin vaiheet ja miten ne liittyvät toisiinsa. Teoriapohjana käytän pääosin Risto Pohjosen (2002) teosta Tietojärjestelmien kehittäminen, jossa hän esittelee näkemyksiään tietojärjestelmien kehitystyön vaiheista ja vaihejakomalleista. Lisäksi esittelen toimeksiantajan JAMES-systeemityömenetelmä, joka toimii toiminnallisen osuuden viitekehyksenä.

Toiminnallisessa osuudessa tavoitteenani oli aloittaa tietohakemiston tekstejä hyödyntävän tietojärjestelmän kehitystyö ja tuottaa siitä tietyt JAMES-menetelmän mukaiset määrittelydokumentit toimeksiantajan käyttöön. Näitä dokumentteja toimeksiantaja voi jatkossa hyödyntää vastaavanlaisissa datan käsittelyyn liittyvissä tietojärjestelmäkehitysprojekteissa. Koska JAMES-menetelmällä toteutettu tietojärjestelmän kehitystyö on yhden henkilön suorittamana aikaa vievä prosessi, päädyimme toimeksiantajan kanssa rajaamaan kehitystyötä siten, että suoritan opinnäytetyössäni vain JAMES-menetelmän Aloitusvaiheen. Muut vaiheet tulen toteuttamaan myöhemmin opinnäytetyöni ulkopuolella.

Edellä mainittujen tavoitteiden lisäksi tavoitteenani oli opinnäytetyöprosessin myötä perehtyä toimeksiantajan järjestelmiin ja käytäntöihin tietojärjestelmäkehityksen osalta. Aloitin työni toimeksiantajaorganisaatiossa juuri ennen opinnäytetyöprosessin alkamista, joten en ollut vielä ehtinyt perehtymään uuden työpaikan järjestelmiin ja käytäntöihin kovin syvällisesti. Opinnäytetyöni aihe liittyi tuleviin työtehtäviini, joten sitä varten tehdyt selvitykset, asioiden opettelu sekä harjoittelu toimivat hyvänä perehdytyksenä työtehtäviin.

Toimeksiantajalla on käytössä erilaisia ohjelmistoja, joiden avulla JAMES-menetelmää toteutetaan organisaation JAVA-teknologiaan liittyvässä sovelluskehityksessä. Tämän opinnäytetyön aikarajoissa minulla ei ollut mahdollisuutta tutustua näihin ohjelmistoihin, joten rajasimme niiden käytön opinnäytetyön ulkopuolelle.

1.2 Opinnäytetyön rakenne

Opinnäytetyöni on toiminnallinen opinnäytetyö, jonka tuotoksena on tietojärjestelmän kehitystyöhön liittyvää dokumentaatiota. Työni teoreettisena viitekehyksenä toimii tietojärjestelmäkehitys, ja tietoperustana käytän erityisesti Risto Pohjosen (2002) teoriaa systeemityöstä, vaihejaosta ja vaihejakomalleista. Työn teoriaosuudessa kerron, millaisia vaiheita systeemityöhön ja tietojärjestelmien kehittämiseen liittyy, ja miten nämä vaiheet liittyvät toisiinsa.

Työn toiminnallisen osuuden aluksi kerron lyhyesti toimeksiantajaorganisaatiosta sekä toimintaympäristöstä. Esittelen myös toimeksiantajaorganisaatiossa käytössä olevan JAMES-systeemityömenetelmän, johon opinnäytetyön toiminnallinen osio perustuu. Seuraavaksi esittelen opinnäytetyön toimeksiannon ja kerron JAMES-systeemityömenetelmän mukaisen tietojärjestelmäkehitysprosessin aloitusvaiheen toteuttamisesta. Kerron, mitä JAMES-lopputuotteita prosessin aikana tuotettiin, ja mitä asioita niiden tuottamiseen liittyi. Toimeksiantajaorganisaatiolle luovutetut JAMES-lopputuotteet on sijoitettu raportin liitteiksi.

Lopuksi esittelen opinnäytetyön johtopäätökset ja kerron, kuinka työni tuloksia on hyödynnetty tai kuinka niitä on tarkoitus hyödyntää organisaatiossa. Pohdiskelen, miten opinnäytetyöprosessi sujui, ja mitä olisi kannattanut ehkä tehdä toisin. Hahmottelen myös joitakin kehitysehdotuksia ja tutkimusideoita, joita voisi tulevaisuudessa toteuttaa. Opinnäytetyössä käytetyt termit määritelmineen esitellään liitteessä 1.

2 Tietojärjestelmien kehittäminen

Organisaatioissa on käytössä erilaisia tietojärjestelmiä. Tietojärjestelmällä tarkoitetaan tässä yhteydessä järjestelmää, joka koostuu laitteista, ohjelmista, tallennetuista tiedoista, järjestelmän käyttäjistä sekä tähän kokonaisuuteen liittyvistä ohjeistuksista. Tietojärjestelmän tarkoituksena on tehostaa, helpottaa tai mahdollistaa jotakin organisaation toimintaa. Yleisiä tavoitteita tietojärjestelmien käytölle ovat esimerkiksi johdon päätöksenteon tukeminen, strategisen kilpailuedun saavuttaminen sekä liike-, perus- ja operatiivisten toimintojen tukeminen. Tietojärjestelmä voi olla joko manuaalinen tai automaattinen. Manuaalinen tietojärjestelmä on esimerkiksi käsin ylläpidettävä kirjanpitojärjestelmä, joka koostuu kuiteista ja kirjanpitolihkosta. Automaattinen tietojärjestelmä puolestaan on yleensä tietotekninen ohjelmistokokonaisuus, joka suorittaa sille määritellyjä tehtäviä. Useimmiten tietojärjestelmä on yhdistelmä sekä manuaalisia että automaattisia toimenpiteitä, jotka täydentävät toisiaan. (Paananen 2003, 364–365.)

Tietojärjestelmät vanhenevat ja lakkaavat vastaamasta organisaation tarpeisiin. Samaan aikaan tekniikka kehittyy ja tarjoaa uusia keinoja ja välineitä järjestelmissä hyödynnettäviksi. Organisaatiot pyrkivätkin tehostamaan ja uudistamaan toimintaansa jatkuvasti tietojärjestelmien kehittämisen avulla. Käytännössä kehittäminen tapahtuu joko parantamalla tai tehostamalla jo käytössä olevien tietojärjestelmien toimintaa, rakentamalla uusia tietojärjestelmiä tai koordinoimalla tietojärjestelmien kehittämistoimenpiteitä ja kehittämisen kokonaissuunnittelua. (Pohjonen 2002, 14–15.)

2.1 Systeemityön vaihejako

Tietojärjestelmien kehitystyön on oltava suunnitelmallista ja systemaattista toimintaa, oli tavoitteena sitten parantaa olemassa olevaa tietojärjestelmää tai rakentaa kokonaan uusi. Tällaisesta systematisoidusta menettelytavasta käytetään termiä systeemityö. Termillä systeemityö viitataan yleisesti tietojärjestelmien ja ohjelmistojen suunnitteluun, ohjelmointiin sekä ylläpitoon. Systeemityössä määritellään kehitystyön tehtäväkokonaisuudet ja toisiaan seuraavat vaiheet. (Pohjonen 2002, 15, 26.) Systeemityön vaihejakoa kutsutaan myös tietojärjestelmän elinkaareksi, joka kuvastaa tietojärjestelmän eri vaiheita olemassaolonsa aikana (Haikala & Märijärvi 2004, 36).

Vaihejako määrittää järjestelmän kehittämisen tehtävät, niiden ajoituksen sekä riippuvuudet toisistaan. Samalla se luo viitekehyksen eri vaiheisiin liittyville käsitteille, ongelmille, menetelmille ja työkaluille. Vaihejaon myötä kehitysprojekteille voidaan määrittää luontevat tarkistuspisteet ja selkiyttää samalla kehitystyön valvontaa ja hallintaa. Kun organisaatiossa noudatetaan sovittua vaihejakoa, kehitysprosessi voidaan standardoida yhdenmukaiseksi ja hyväksi havaitut käytännöt saadaan yleiseen käyttöön. (Pohjonen 2002, 26.)

Jokaisen vaiheen tehtävät ja dokumentaatiot on suoritettava huolella, sillä kunkin vaiheen tulos toimii seuraavan vaiheen syötteenä. Lisäksi jokaista vaihetta tukevat koko kehitysprosessin läpileikkaavat toiminnot, joita ovat esimerkiksi vaatimustenhallinta, laadunvarmistus, dokumentaatio sekä riskienhallinta. Systeemityön vaihejakoon liittyvistä vaiheista on käytössä erilaisia nimityksiä ja kokoonpanoja, joista tässä opinnäytetyössä esitellään Pohjosen (2002) näkemys. Hänen mukaansa systeemityön vaihejako koostuu esitutkimuksesta, vaatimusmäärittelystä, järjestelmäanalyysistä, suunnittelusta, toteutuksesta, testauksesta, käyttöönotosta ja ylläpidosta. (Pohjonen 2002, 26, 39.)

Esitutkimusvaihe aloitetaan, kun organisaatiossa on tunnistettu tarve kehittää jo olemassa olevaa tietojärjestelmää tai tuottaa aivan uusi tietojärjestelmä. Esitutkimusvaiheen tavoitteena on selvittää, onko kehityshankkeen toteuttaminen mahdollista ja mielekästä. Tässä vaiheessa ei tehdä mitään teknisiä ratkaisuja tai suunnitelmia lopputuoteseen liittyen. Vaiheen aikana selvitetään, miksi uusi tietojärjestelmä tulisi rakentaa, mitä tavoitteita sille asetetaan, mitä viiteryhmiä järjestelmällä on, ja mitä eri vaihtoehtoja on olemassa sen toteuttamiselle. (Pohjonen 2002, 27.) Haikala ja Märijärvi (2004, 37) korostavat, että esitutkimusvaihe on tietojärjestelmän elinkaaren tärkein vaihe, sillä huolella kartoitetut asiakasvaatimukset ovat edellytys toimivan ja laadukkaan tietojärjestelmän kehittämiseksi.

Pohjosen (2002, 27) mukaan esitutkimusvaiheen lopputuotoksena on esitutkimusraportti. Sen sisältämien tietojen perusteella johtava taho voi tehdä päätöksen tietojärjestelmähankkeen aloittamisesta tai hylkäämisestä. Raportista tulisi ilmetä ainakin organisaation nykytilanne kyseisen kehityshankkeen osalta, kuvaukset ongelmista, jotka uuden

tietojärjestelmän tulisi ratkaista sekä kuvaukset niistä sidos- ja viiteryhmistä, joita kehityshanke koskee. Lisäksi raporttiin on hyvä sisällyttää järjestelmälle asetetut alustavat tavoitteet ja rajaukset, toimintavaihtoehtojen kuvaukset sekä alustava suunnitelma, jonka avulla kehittämishanke on tarkoitus toteuttaa. Esitutkimusraportti ottaa siis kantaa siihen, miksi kyseinen tietojärjestelmä tulisi rakentaa tai olla rakentamatta (Haikala & Märijärvi 2004, 37).

Jos esitutkimusraportin perusteella on päädytty siihen, että kehittämishanke käynnistetään, siirrytään **vaatimusmäärittelyvaiheeseen**. Vaatimusmäärittelyssä muodostetaan käsitys siitä, mitä kohdejärjestelmältä odotetaan. Vaiheen tarkoituksena on selvittää ja dokumentoida sidos- ja viiteryhmien järjestelmälle asettamat vaatimukset. Selvitystyötä tehdään usein joko haastatteluiden, ideointipalaverien tai markkinatutkimusten avulla. Olennaista on se, että vaatimusten tulisi olla loppuun asti harkittuja, realistisia ja keskenään ristiriidattomia. Huomiota tulisi kiinnittää myös siihen, että vaatimusten toteutumisesta on voitava arvioida ja mitata. Liian epämääräisten tai vaikeaselitteisten vaatimusten toteutumista on vaikeaa todentaa. (Pohjonen 2002, 28–29.)

Vaatimukset voivat olla toiminnallisia tai ei-toiminnallisia. Toiminnalliset vaatimukset kertovat, mitä järjestelmän odotetaan tekevän, esimerkiksi: toimihenkilön tulee voida selata, lisätä, muokata ja poistaa tietohakemiston tietokuvauksia. Ei-toiminnalliset vaatimukset ottavat kantaa reunaehtoihin, joiden vallitessa järjestelmä täyttää toiminnalliset vaatimuksensa. Ei-toiminnallinen vaatimus on esimerkiksi: tietohakemistosovelluksen tulee toimia Internet Explorer- ja Mozilla Firefox -selaimilla. Lisäksi voidaan määritellä tiettyjä rajoitteita, jotka ovat järjestelmälle määriteltyjen toiminnallisten vaatimusten rajoituksia, esimerkiksi: toimihenkilö tarvitsee pääkäyttäjän oikeudet, jotta hän voi muokata tai poistaa tietokuvauksia. Vaatimukset ja rajoitteet on syytä priorisoida. Tällöin kaikille osapuolille on selvää, mitkä ominaisuudet tullaan toteuttamaan ja mitkä jätetään huomiotta tai toteutetaan tulevissa versioissa. (Pohjonen 2002; 28, 31.)

Vaatimusmäärittelyvaiheen lopputuloksena on vaatimusmäärittelydokumentti. Dokumentissa tulee esitellä kehittämishankkeen toimeksianto ja antaa yleiskatsaus organisaation nykytilanteesta kohdejärjestelmän osalta. Samalla on kuvattava kohdejärjestelmän

ja sille asetettujen tavoitteiden pääpiirteet. Lisäksi vaatimusmäärittelydokumenttiin on kirjattava jokaisen toiminnallisen ja ei-toiminnallisen vaatimuksen kuvaus, jokaisen rajoitteen kuvaus, vaatimukset ja rajoitteet numeroituina ja priorisoituina sekä mahdolliset asiaan liittyvät lisäselvitykset. (Pohjonen 2002, 30–31.) Käytännössä kyse on siitä, että kootuista asiakasvaatimuksista on muokattu tietojärjestelmän täsmälliset ohjelmistovaatimukset (Haikala & Märijärvi 2004, 39).

Seuraavaksi on vuorossa järjestelmän määrittely eli **järjestelmäanalyysi**. Vaiheen tarkoituksena on selvittää, mitä kohdejärjestelmän tulee tehdä. Tämä tapahtuu siten, että vaatimusmäärittelyvaiheessa tunnistetut vaatimukset analysoidaan ja niistä johdetaan toteutustavasta riippumaton toiminnallinen määrittely. Järjestelmän toiminnallinen määrittely sisältää dokumentteja, joista selviää yleiskuvaus rakennettavan järjestelmän tarkoituksesta ja sen toiminnasta yleisellä tasolla. Toiminnallisessa määrittelyssä kuvataan myös järjestelmän toimintaympäristö ja käyttäjät. Lisäksi järjestelmän yleiset rajoitteet ja sen käyttöön liittyvät oletukset ja riippuvuudet dokumentoidaan. Kaikki järjestelmään liittyvät toiminnot tulee kuvata yksityiskohtaisesti, samoin järjestelmässä käsiteltävien tietojen ja tietokantojen rakenteet. Toiminnallisen määrittelyn dokumentaatio sisältää myös järjestelmän suorituskyykyyn, käytettävyyteen, virhetilanteista toipumiseen sekä turvallisuuteen liittyviä määrittelyitä. (Pohjonen 2002, 31–32.)

Määrittelyiden jälkeen alkaa **suunnitteluvaihe**. Sen aikana tehdään tarkkoja suunnitelmia järjestelmän toteuttamiseksi. Tässä vaiheessa järjestelmän toiminnallinen määrittely muunnetaan tekniseksi määrittelyksi, jossa kuvataan järjestelmän toteutus. Käytännössä tämä tapahtuu yleensä arkkitehtuuri- ja moduulisuunnittelun avulla. Arkkitehtuurisuunnittelussa järjestelmän rakenne ositetaan mahdollisimman pieniin osiin, moduuleihin, jotka voidaan antaa eri tahoille suunniteltaviksi ja toteutettaviksi. (Haikala & Märijärvi 2004, 40). Moduulilla tarkoitetaan kehitettävään järjestelmään liittyvää, mahdollisimman itsenäistä osakokonaisuutta, joka kommunikoi ympäristönsä kanssa tiettyjen rajapintojen kautta. Moduulisuunnittelussa suunnitellaan jokaisen yksittäisen moduulin sisäinen rakenne. Molemmissa suunnittelulinjoissa tulisi pyrkiä selkeyteen, ymmärrettävyyteen, tehokkuuteen, luotettavuuteen, ylläpidettävyyteen sekä siirrettävyyteen. (Pohjonen 2002, 32–33.)

Suunnitteluvaiheen tuloksena on tekninen määrittely. Sen tulee sisältää ainakin seuraavat dokumentaatiot: tiivistelmä järjestelmän tarkoituksesta, kuvaus järjestelmän sovel-
lusalueesta ja järjestelmän osuudesta siinä sekä kuvaus järjestelmän laitteisto- ja ohjel-
mistoympäristöstä. Lisäksi dokumentaatiosta tulisi selvittää järjestelmän toteutuksen reu-
naehdot, järjestelmän ja sen ympäristön vuorovaikutus, järjestelmän yleiset ratkaisupe-
riaatteet, ohjelmisto- ja tietokanta-arkkitehtuuri sekä jokaisen yksittäisen moduulin ku-
vaus. Myös mahdolliset tekniset erityisratkaisut sekä vaihtoehtoiset tai hylätyt ratkaisut
on syytä dokumentoida jo tässä vaiheessa. (Pohjonen 2002, 33.)

Suunnitteluvaiheen jälkeen siirrytään **toteutusvaiheeseen**. Tässä vaiheessa rakennetta-
van järjestelmän ohjelmisto tai sen osamoduuli toteutetaan jollakin ohjelmointikielellä
tai sovelluskehittimellä. Toteutusta tukevan työvälineen valintaan vaikuttavat erilaiset
tekijät. Valintaa tehtäessä on huomioitava, mistä sovellusalueesta on kyse, millaisia oh-
jelmistotuotannon malleja tai menetelmiä käytetään, mitkä ovat tehokkuusvaatimukset,
ja millainen toteutus- ja käyttöympäristö on kyseessä. Mikäli toteutus tehdään osissa,
suoritetaan vaiheen lopussa osamoduulien integrointi toimivaksi järjestelmäksi. (Pohjo-
nen 2002, 34.) Haikala ja Märijärvi (2004, 40) korostavat toteutusvaiheen päättyvän
siihen, kun tietojärjestelmän toteuttajat ovat saaneet tuotettua ensimmäisen virheettö-
män käännöksen, minkä jälkeen siirrytään seuraavaan vaiheeseen.

Koko toteutusvaiheen ajan on huolehdittava siitä, että toteutus vastaa edellisissä vai-
heissa tunnistettuja vaatimuksia ja määrittelyitä. Lisäksi tulee tarkkailla aktiivisesti mah-
dollisia uusia esille nousevia vaatimuksia ja reagoida niihin ennalta sovitulla tavalla.
Tarpeen mukaan toteutustyö voidaan keskeyttää ja palata aikaisempiin vaiheisiin tar-
kentamaan esimerkiksi vaatimusmäärittelyä tai suunnittelua. (Pohjonen 2002; 34, 39.)

Toteutusvaiheen haasteita ovat pyrkimys siirrettävyyteen ja ylläpidettävyyteen. Usein
rakennettavan järjestelmän odotetaan toimivan erilaisissa laitteissa tai käyttöjärjestel-
missä, mikä on huomioitava sekä toteutusvaiheen työkaluvalinnoissa että ohjelmoinnis-
sa. Järjestelmien on usein tarkoitus olla varsin pitkäikäisiä, ja siksi ohjelmarakenteiden ja
ohjelmointikoodin tulisi olla selkeitä, ymmärrettäviä ja helposti ylläpidettäviä. Ohjel-
moinnissa tulisikin noudattaa yhteiseksi sovittuja standardeja ja nimeämiskäytäntöjä.

Lisäksi ohjelmakoodia tulee dokumentoida ja kommentoida siten, että muutkin pystyvät ymmärtämään ja muokkaamaan sitä. (Pohjonen 2002, 35.)

Testausvaiheessa uusi järjestelmä testataan mahdollisten virheiden varalta. Ideaalitalanteessa kattavan testauksen avulla löydettäisiin kaikki järjestelmän virheet ja puutteet ennen käyttöönottoa, mutta käytännössä aikataulu- ja budjettipaineet johtavat usein testausvaiheen typistämiseen. Tällöin voidaan havaita ja korjata osa virheistä, mutta täyttä virheettömyyttä ei voida taata. Järjestelmän kehitystyö jatkuu usein vielä käyttöönottovaiheen jälkeen. (Pohjonen 2002, 35–36.)

Testauksessa käytetään usein niin sanottua V-mallia, jossa testausta suoritetaan niin moduulien, integroinnin kuin järjestelmänkin osalta. Moduulitestauksen tarkoituksena on löytää virheitä yksittäisistä moduuleista, integrointitestauksessa keskitytään moduulien yhteistoimintaan ja järjestelmätestauksessa testataan koko järjestelmän suorituskykyä ja toimintaa. V-mallin mukainen järjestelmätestaus suunnitellaan jo osana ohjelmiston analyysiä, ja itse testaus suoritetaan vertaamalla valmista järjestelmää sen toiminnalliseen määrittelyyn. Integrointitestaus taas suunnitellaan arkkitehtuurisuunnittelun yhteydessä, ja moduulitestaus moduulisuunnittelun yhteydessä. Haikala & Märijärvi 2004, 40.) Olennaista on se, että testaus ja testiaineistot perustuvat määrittely- ja suunnitteluvaiheen tuloksiin, eivät laadittuun ohjelmakoodiin (Pohjonen 2002, 36).

Kun järjestelmä on testattu ja testauksessa havaitut virheet on korjattu, voidaan siirtyä **käyttöönottovaiheeseen**. Siinä järjestelmä viedään tuotantoon suorittamaan sille määriteltyjä tehtäviä. Ennen varsinaista käyttöönottoa on syytä tehdä suunnittelutyötä ja valmisteluja, jotta itse käyttöönotto sujuisi mahdollisimman hyvin. Esimerkiksi olemassa olevat tiedot, tietokannat ja tiedostot on siirrettävä uuteen järjestelmään, aikaisemmat tai rinnakkaiset järjestelmät huomioitava, ja käyttäjäkunta sekä ylläpitohenkilökunta koulutettava uuden järjestelmän käyttöön. Lisäksi on huolehdittava tarvittavien tilojen valmistelusta sekä laitteiden ja ohjelmistojen asennuksista. Tämän vaiheen aikana on syytä sopia myös selkeät ylläpitovastuut, joiden perusteella nimetyt tahot huolehtivat uuden järjestelmän toiminnasta ja kehittämisestä. (Pohjonen 2002, 37.)

Tietojärjestelmän viimeinen ja samalla pisin vaihe on **ylläpitovaihe**, joka jatkuu aina järjestelmän elinkaaren loppuun asti. Tämän vaiheen tarkoituksena on huolehtia tuotantokäytössä olevan järjestelmän toiminnasta tarvittavilla virheiden korjauksilla, jatkokehityksellä ja mahdollisilla muutostoimenpiteillä. Ylläpitovaiheessa voidaan tehdä erilaisia ylläpitotoimia. Korjaavaa ylläpitoa on käyttöönoton jälkeen havaittujen virheiden korjaaminen. Sopeuttavaa ylläpitoa tehdään, kun järjestelmä siirretään uusiin ympäristöihin. Lisäksi saatetaan suorittaa täydentävää ylläpitoa, jolla tarkoitetaan uusien ominaisuuksien kehittämistä. Ennakoivaa ylläpitoa on järjestelmän tai sen dokumentaation tason parantaminen tulevia ylläpitotilanteita varten. (Pohjonen 2002, 37.)

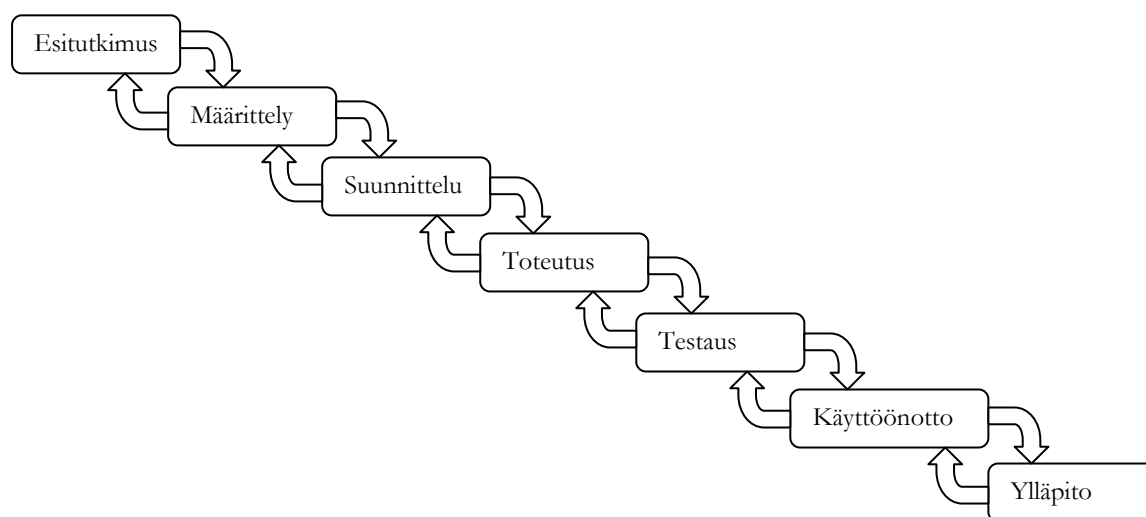
Ylläpitovaiheessa tärkein tekijä on kattava dokumentaatio, jotta järjestelmän taustalla vaikuttavat suunnittelu- ja toteutusratkaisut olisivat ymmärrettäviä ja jäljitettäviä. Dokumentointityön on jatkuttava myös ylläpitovaiheessa, jolloin kaikki järjestelmään tehdyt lisäykset ja muutokset on dokumentoitava standardimuotoisena. Motivaatiota ylläpidettävyyden helpottamiseen ja järjestelmän dokumentointiin saattaa lisätä tieto siitä, että järjestelmään sen elinkaaren aikana panostettavista resursseista 70 % kuluu ylläpitoon. (Pohjonen 2002, 38.)

2.2 Systeemityön vaihejakomallit

Tietojärjestelmän kehittämiseen liittyvien vaiheiden keskinäinen järjestys ja vaiheiden ajoitus saattaa vaihdella organisaatiosta tai ohjelmistoprojektista riippuen. Vaiheiden järjestystä ja ajoitusta on pyritty mallintamaan vaihejakomallien eli elinkaarimallien avulla. Vaihejakomalleja on olemassa useita erilaisia. Tässä opinnäytetyössä esitellään kolme yleisesti käytössä olevia vaihejakomallia: vesiputousmalli, prototyypilähestymistapa sekä spiraalimalli. Vaihejakomallit ovat suuntaa-antavia kuvauksia kehitystyön etenemisestä ideaalitapauksessa, eikä niitä aina ole mahdollista tai tarkoituksenmukaista noudattaa kirjaimellisesti. Usein organisaatiolla onkin sen omiin tarpeisiin räätälöity vaihejakomalli, jota sovelletaan organisaation sisäisiin ohjelmistoprojekteihin. (Pohjonen 2002, 39.)

Vesiputousmalli (Kuvio 1) on vanhin mutta samalla yleisin käytössä oleva vaihejakomalli. Sen kehitys alkoi jo 1960-luvulla, kun ensimmäisten suurten tietokoneohjelmien

rakentamisprojekteissa tarvittiin systemaattisia työprosesseja. Vesiputousmallin rakenne muistuttaa vesiputousta, jossa vaiheet esitutkimuksesta ylläpitoon seuraavat toisiaan tietyssä järjestyksessä. Puhtaassa vesiputousmallissa uusi vaihe aloitetaan edellisen päätyttyä, ja prosessi etenee pääsääntöisesti eteenpäin. Taaksepäin peruuttamista pyritään välttämään hiomalla jokainen vaihe kerralla loppuun saakka valmiiksi, jotta tarvetta palaamiseen ei tulisi. (Haikala & Mikkonen 2011, 36–37; Pohjonen 2002, 40.)



Kuvio 1. Vesiputousmalli Pohjosen (2002, 40) mukaan

Käytännössä tietojärjestelmien kehitysprosessi etenee vesiputousmallin mukaan suora-
viivaisesti hyvin harvoin. Vaikka vaatimusmäärittely- ja suunnitteluvaiheet tehtäisiin
erittäin huolellisesti, voi toteutus- tai testausvaiheessa tulla ilmi virheitä tai puutteita,
joiden korjaaminen vaatii palaamista aikaisempiin vaiheisiin. Esimerkiksi lopputuotetta
voidaan esitellä asiakkaalle vasta prosessin loppuvaiheessa, kun kaikki vaiheet on käyty
läpi. Mikäli asiakas ei olekaan tyytyväinen tuotteeseen ja haluaa siihen muutoksia, pro-
sessi joudutaan aloittamaan alusta. Vesiputousmallin mukaan kunkin vaiheen tuotokset
toimivat seuraavan vaiheen syöteinä, joten käytännössä koko prosessi joudutaan uusi-
maan palaamiskohdasta alkaen, mikä pidentää kehitysohjelmia ja aiheuttaa merkittäviä
kustannuksia. (Pohjonen 2002, 40.)

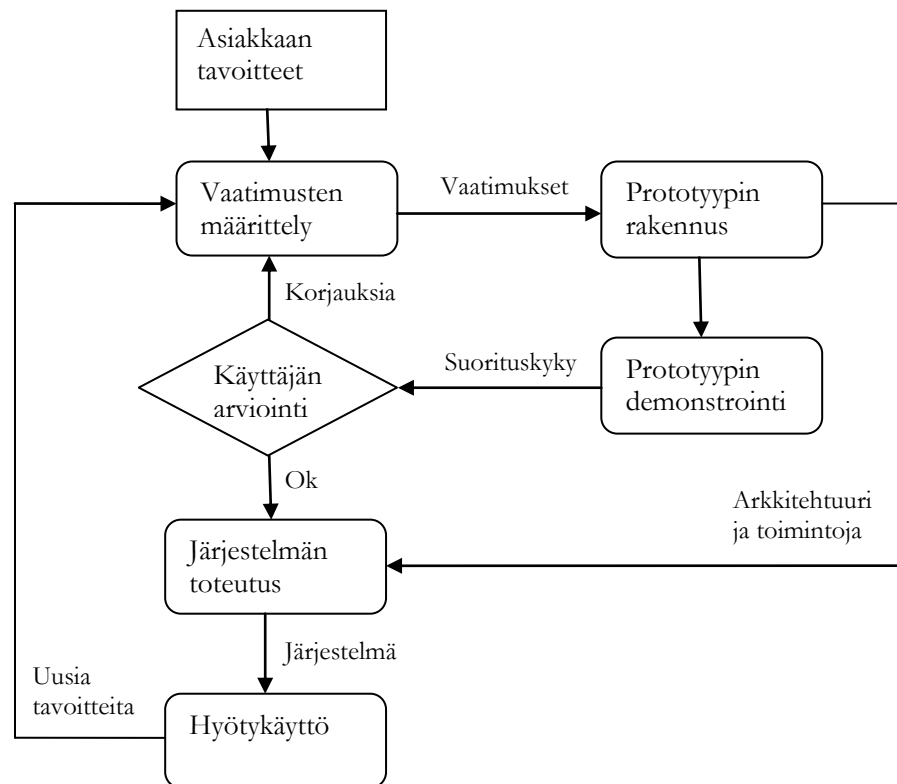
Koska vesiputousmalli on osoittautunut vaikeaksi noudattaa sellaisenaan, sitä käytetään
tietojärjestelmien kehitysohjelmia yleensä soveltaen. Haikala ja Mikkonen (2011, 36–

37) viittaavat vesiputousmallia 1970-luvulla kehittäneen Roycen artikkeliin. Siinä korostetaan kehitysprosessissa taaksepäin palaamisen mielekkyyttä ja jopa tarpeellisuutta mahdollisten ongelmien havaitsemiseksi sekä kustannusten leikkaamiseksi. Roycen ajatusten myötä tietojärjestelmien kehitysprojekteihin tuli mukaan iteratiivinen lähestymistapa. Iteratiivisessa kehitysprojektissa voidaan tarpeen mukaan palata prosessissa joustavasti taaksepäin määrittelyiden tarkentamiseksi ja virheiden korjaamiseksi mahdollisimman varhaisessa vaiheessa. Kuviossa 1 ajatusta iteratiivisuudesta kuvastavat vaiheiden välillä olevat nuolet. Iteratiivisen lähestymistavan pohjalta on myös kehitetty kokonaan uusia vaihejakomalleja, jotka ovat perinteistä vesiputousmallia joustavampia. Kriittikistä huolimatta vesiputousmalli on ollut merkittävä askel kohti modernia ohjelmistokehitystä, ja useimpien vaihejakomallien rakenteesta onkin edelleen tunnistettavissa vesiputousmallin elementtejä. (Haikala & Mikkonen 2011, 36–37.)

Toinen yleinen vaihejakomalli on **prototyypilähestymistapa**. Malli on toimiva esimerkiksi silloin, kun tietojärjestelmän tilaaja ei aluksi itseään tiedä aivan tarkkaan, millainen tietojärjestelmän tulisi olla tai miten sen tulisi toimia. Tällaisessa tapauksessa olisi kovin riskialtista toteuttaa kehitysprojekti vesiputousmallin mukaisesti, jolloin tilaaja saisi valmiin tuotteen nähtäväkseen vasta projektin loppuvaiheessa. Mikäli hän ei olisi tyytyväinen lopputulokseen tai haluaisi järjestelmään vielä uusia ominaisuuksia, jouduttaisiin koko vesiputousprosessi aloittamaan alusta. Prototyypilähestymistapaa noudattaessa asiakkaalle tuotetaan hyvin varhaisessa vaiheessa arvioitavaksi järjestelmän prototyyppi, jonka perusteella hän voi tarkentaa vaatimuksiaan. (Paananen 2003, 372–373.)

Prototyypilähestymistavan (Kuvio 2) mukainen tietojärjestelmien kehitysprojekti aloitetaan selvittämällä ja analysoimalla asiakkaan vaatimukset ja tavoitteet. Analyysin perusteella rakennetaan ensimmäinen epätäydellinen prototyyppi, jonka avulla asiakkaalle voidaan demonstroida kohdejärjestelmän yleistä toiminnallisuutta. Asiakkaan palautteen perusteella prototyyppiä parannellaan, minkä jälkeen seuraa uusi demonstraatio. Tätä parantelu - demonstraatio -kiertoa toistetaan niin kauan, että asiakas on tyytyväinen prototyyppiin. Lopuksi kohdejärjestelmä toteutetaan prototyypin pohjalta. Proto-

tyyppivaiheessa valmistuneita määrittelyjä, toimintoja ja arkkitehtuuriratkaisuja hyödynnetään toteutusvaiheessa. (Pohjonen 2002, 41.)



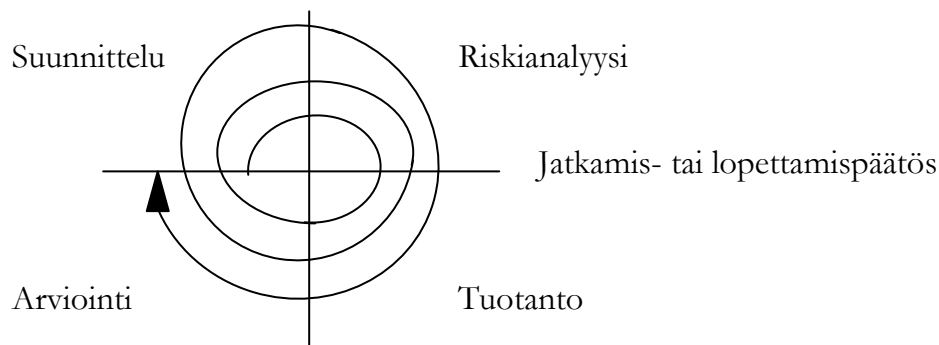
Kuvio 2. Prototyypilähestymistapa Pohjosen (2002, 41) mukaan

Prototyypilähestymistavassa voidaan erottaa kaksi eri päätyyppiä; evoluutioprototyyppi ja poisheitettävä prototyyppi. Evoluutioprototyypillä tarkoitetaan sellaista prototyyppiä, jota kehitetään prosessin aikana vähitellen valmiiksi tuotteeksi. Poisheitettävä prototyyppi on taas sellainen prototyyppi, jota käytetään usein järjestelmän mallintamiseen. Esimerkiksi järjestelmän käyttöliittymän hahmottelussa paperipiirros toimii poisheitettävänä prototyypinä, ja varsinainen toteutustyö tehdään jollakin muulla välineellä. (Haikala & Mikkonen 2011, 38.)

Pohjonen (2002, 42) kritisoi prototyypilähestymistapaa siitä, että sen mukaisessa tietojärjestelmäkehitysprojektissa joudutaan rakentamaan järjestelmä kahteen kertaan; ensin prototyypin rakennusvaiheessa ja lopuksi varsinaisen järjestelmän toteutusvaiheessa. Lisäksi prototyypin pelkistetty rakenne voi johtaa siihen, että osa ongelmista ja virheistä tulee esille vasta myöhemmässä vaiheessa. Haikala ja Mikkonen (2011, 39) varoittavat

lisäksi siitä, että asiakas saattaa luulla prototyypin olevan jo valmis järjestelmä. Tämä saattaa johtaa kiristyvään aikatauluun, jolloin houkutus hyödyntää ei-valmiita komponentteja kasvaa riskeistä huolimatta.

Kolmas tällä hetkellä hyvin yleisesti käytössä oleva vaihejakomalli on **spiraalimalli**. Spiraalimalli (Kuvio 3) koostuu neljästä vaiheesta, jotka ovat suunnittelu-, riskianalyysi-, tuotanto- ja arviointivaihe. Suunnitteluvaiheessa määritellään tavoitteet, vaihtoehdot ja rajoitukset asiakkaan toiveiden perusteella. Riskianalyysivaiheessa arvioidaan eri vaihtoehtoihin liittyviä ongelmia ja riskejä sekä tehdään projektin jatkamis- tai lopettamispäätös. Tuotantovaiheessa valmistetaan kyseisen järjestelmävaiheen tuotos, ja arviointivaiheessa asiakas arvioi tätä tuotosta. Mallille on ominaista ajatus iteratiivisuudesta, eli vaiheita toistetaan uudelleen ja uudelleen jatkuvasti tarkentaen niin kauan, että järjestelmä on valmis. (Pohjonen 2002, 42.)



Kuvio 3. Spiraalimalli Pohjosen (2002, 42) mukaan

Spiraalimallin joustavuutta lisää se, ettei siinä kiinnitetä tuotantovaiheen menetelmiä, vaan se antaa vapauden soveltaa esimerkiksi vesiputous- tai prototyyppimallia. Jatkuvan riskianalyysin avulla saadaan tietoa projektin etenemisedellytyksistä, ja toisaalta myös projektin lopettamispäätös voidaan tehdä nopeasti tilanteen niin vaatiessa. Spiraalimallin käyttämisessä on omat ongelmansa; mallin toteuttaminen vaatii asiakkailta aktiivisuutta ja toimijoilta riskianalyysin hallitsemista. Lisäksi vaiheiden iteratiivinen toistaminen useaan kertaan saattaa viedä paljon aikaa. (Pohjonen 2002, 43.)

Eräs spiraalimallin iteratiivista ajattelua noudattava ja suosittu vaihejakomalli on IBM:n kehittämä ohjelmistokehityksen prosessikehikko RUP eli Rational Unified Process.

Siinä kehitystyö on jaettu neljään päävaiheeseen, jotka ovat inception, elaboration, construction ja transition. Jokaiselle vaiheelle on määritelty omat tuloksensa, jotka vaiheen aikana tulisi saavuttaa. Esimerkiksi inception-vaiheessa tulisi selvittää asiakkaiden esittämät vaatimukset kehitettävälle järjestelmälle. Vaiheissa suoritetaan n-määrä iteraatioita. Jokaisessa iteraatiossa suoritetaan vesiputous vaatimusmäärittelystä suunnitteluun, toteutukseen ja testaukseen. (Haikala & Märijärvi 2004, 45–46).

3 Toimeksiantajaorganisaatio

Opinnäytetyön toimeksiantajaorganisaatio on Kansaneläkelaitos eli lyhyemmin Kela. Kela on itsenäinen julkisoikeudellinen laitos, joka huolehtii Suomessa asuvien ihmisten perusturvasta. Kela perustettiin vuonna 1937 huolehtimaan alun perin nimensä mukaisesti kansaneläkkeiden maksuista. Sitten toiminta on laajentunut käsittämään kaiken ikäisten kansalaisten perusturvan vauvasta vaariin. Erilaisia etuuksia on nykyään noin 100, muun muassa lapsilisä, opintotuki, asumistuki, sotilasavustus ja kansaneläke. Erilaisia asiointitapahtumia on vuosittain yli 12 miljoonaa, ja etuushakemuksia käsitellään vuosittain yli neljä miljoonaa. (Kela 2012a.) Vuonna 2011 Kelan etuuskulut olivat 12,5 mrd. euroa (Kela 2012b).

Etuuskäsittelytyön ohella Kelan toimenkuvaan kuuluu myös tiedottaminen, tilastoiminen ja tutkiminen. Tarkoituksena on tuottaa tietoa päätöksenteon tueksi sekä informoida kansalaisia ja muita tahoja etuuksista ja niiden saamisehdoista. (Kela 2012a.) Hyvänä esimerkkinä tilastoinnista toimii vuodesta 1965 alkaen julkaistu Kelan tilastollinen vuosikirja, johon tilastoidaan vuosittain luvut ja tiedot Kelan hoitamista etuuksista sekä niiden saamisen edellytykset. Kelan tietojärjestelmiin tallennettuja tietoja hyödyntävät Kelan omien tutkijoiden lisäksi tietyt ulkoiset tahot, jotka tilaavat Kelasta tutkimusaineistoa. (Kela 2012c.)

3.1 Organisaation esittely

Kelalla on yli 200 toimipistettä ympäri Suomea. Päätoimitalo on Helsingin keskustassa Nordenskiöldinkadulla, muu keskushallinto sijoittuu Pitäjänmäelle, Jyväskylään ja Turkuun. Vuoden 2011 lopussa Kelassa työskenteli 6146 eri alojen ammattilaista. Pääosa henkilöstöstä työskentelee asiakaspalvelussa ja etuuksien ratkaisussa. Lisäksi erilaisissa asiantuntija-, suunnittelu- ja johtotehtävissä työskentelee mm. juristeja, farmaseutteja, lääkäreitä ja IT-alan ammattilaisia. (Kela 2012b.)

Opinnäytetyön toimeksiantaja työskentelee Kelan IT-osastolla. Tietojenkäsittelyllä ja IT-osaamisella on Kelassa pitkät perinteet: kansaneläketietoja alettiin käsitellä reikäkortteilla jo 1940-luvulla. Varsinainen tietojenkäsittely alkoi 1960-luvulla, kun siirryttiin au-

tomaattiseen eläkkeiden maksatusjärjestelmään. Etuusjärjestelmiä on kehitetty vuosien saatossa tarpeiden mukaan. Nyt työn alla on etuusjärjestelmien uudistamishanke Arkki, jonka tavoitteena on uudistaa kaikki Kelan 40 etuustietojärjestelmää sekä teknisesti että toiminnallisesti. (Kelanetti 2011, Kela 2012b.)

Etuuksiin liittyvät IT-palvelut pyritään tuottamaan IT-osastolla itse sen sijaan, että ne ostettaisiin ulkopuolelta. IT-osaston toiminta-ajatuksena on suunnitella, ylläpitää ja kehittää tietojärjestelmiä sekä tuottaa, toimittaa ja hankkia Kelan käyttämät tieto- ja viestintäteknologiajärjestelmät. Tällä hetkellä IT-osasto työllistää yli 450 tietotekniikan ammattilaista neljässä eri toimipisteessä: Päätoimitalolla Nordenskiöldinkadulla, Pitäjänmäessä, Jyväskylässä sekä Turussa. (Kelanetti 2012.)

Organisatorisesti IT-osasto on jaettu seitsemään eri ryhmään, jotka ovat erikoistuneet oman sektorinsa mukaisiin toimintoihin. (Kelanetti 2012.) Toimeksiantajani edustaja ja tämän opinnäytetyön ohjaaja toimii Kelan IT-osaston Järjestelmäkehitysryhmän Sovelluskehitystiimi 4:n kehittämispäällikkönä. Hän osallistuu tiiminsä kanssa Kelan tilastointi- ja raportointijärjestelmien ylläpitoon ja kehittämiseen. Lisäksi hän on ollut mukana Kelan tietoarkkitehtuurin kehittämisessä. (Rinta-Paavola 2012.)

3.2 Toimintaympäristön esittely

Työstin opinnäytetyötä sekä työpaikallani Kelan IT-osastolla että kotonani. Sidosryhmien haastattelut, tuotettujen dokumenttien katselmoinnit sekä toimeksiantajani kanssa käydyt keskustelut tapahtuivat työpaikalla. JAMES-dokumentit sekä opinnäytetyön kirjallisen osuuden kirjoitin kotona. JAMES-menetelmän käyttöä opastava JAMES-käsikirja ja menetelmän mukaiset dokumenttipohjat sijaitsevat Kelan palvelimella, josta sain tallennettua ne omalle Windows 7-työasemalleni tarkastelua ja muokkausta varten. Opinnäytetyön liitteiksi sijoitetut JAMES-lopputuotedokumentit luotiin MsWord2007-tekstinkäsittelyohjelmalla olemassa olevien mallipohjien mukaisiksi.

Kehitettävä tietojärjestelmä tulee toimimaan Kelan Mainframe-ympäristössä. Tilastointi- ja raportointitehtäviä suorittavat henkilöt tulevat käyttämään järjestelmää, siihen liit-

tyvää makro-ohjelmaa sekä makro-ohjelman avulla täydennettäviä tiedostoja omilta työasemiltaan. Kehitettävä järjestelmä tullaan toteuttamaan SAS-ohjelmointikielellä.

4 Toimeksiantajan JAMES-systeemityömenetelmä

Opinnäytetyön toimeksiantajan ohjelmisto- ja tietojärjestelmäkehityksen vaihejakomallina ja systeemityömenetelmänä on ollut aikaisemmin käytössä perinteinen vesiputousmalli. Kuten työn teoriaosuudessa todettiin, vesiputousmallin periaatteisiin kuuluu, että määrittely-, suunnittelu-, toteutus- ja käyttöönottovaiheet seuraavat toisiaan kronologisessa järjestyksessä. Tällöin uusi vaihe aloitetaan vasta sitten, kun edellinen vaihe tehtävienne on saatu päätökseen. Savolaisen (2011) mukaan vesiputousmalli on ollut toimiva menetelmä ja kattanut toimeksiantajan ohjelmistokehityksen tarpeet viime vuosiin saakka. Tietojärjestelmien ja toimintaympäristöjen monimutkaistuminen sekä asteittainen siirtyminen JAVA-ohjelmointikielen ja -teknologian käyttöön ovat kuitenkin luo- neet tarvetta joustavampaan ohjelmistokehitykseen.

Vastauksena tähän tarpeeseen toimeksiantajan menetelmätiimi on kehittänyt JAMES-systeemityömenetelmän. JAMES on lyhenne sanoista JAva-MEnetelmä-Systeemityö, ja se on kehitetty toimeksiantajaorganisaatiossa alun perin erityisesti JAVA-ohjelmointikielellä suoritettavalle ohjelmistokehitykselle. Koska menetelmässä ei kuitenkaan ole mitään erityisiä sidoksia JAVA-teknologiaan, se soveltuu käytettäväksi myös esimerkiksi SAS- ja PL/1-ohjelmistoihin liittyvien järjestelmien toteutuksessa. Toistaiseksi JAMES-menetelmää on kuitenkin hyödynnetty vain JAVA-teknologiaan liittyvissä projekteissa, ja muilla teknologioilla toteutetut projektit on työستetty vesiputousmallia hyödyntäen. (Savolainen, 2011.)

JAMES-menetelmän avulla tietojärjestelmän kehitysprosessin elinkaarivaiheiden työ- vaiheet, tehtävät, niitä suorittavat roolit sekä syntyvät lopputulokset kuvataan ja doku- mentoidaan kattavasti. Tällainen systemaattinen toiminta tuottaa määrämuotoisia ja samansisältöisiä dokumentteja, jotka helpottavat paitsi siirtymistä seuraaviin työvaihei- siin, myös valmistuvan järjestelmän ylläpitoa ja jatkokehittämistä. JAMES perustuu RUP:iin (Rational Unified Process), joka on hyväksi havaituista käytännöistä koostettu ohjelmistokehityksen prosessikehy. (Savolainen, 2011.) Teoriaosuudessa esitellyistä vaihejakomalleista JAMES-menetelmän toiminta-ajatus noudattelee lähinnä Pohjosen (2002, 42) esittelemää spiraalimallia, jota havainnollistettiin kuviossa 3.

Toimeksiantajalla on käytössä erilaisia ohjelmistoja, joiden avulla JAMES-menetelmää toteutetaan organisaatiossa. Ohjelmistoja käytetään esimerkiksi vaatimusmäärittelyn tukena sekä kaavioiden mallinnuksessa. Tämän opinnäytetyön aikarajoissa ei ollut mahdollista tutustua näihin ohjelmistoihin, ja ne rajattiin opinnäytetyön ulkopuolelle. Näitä apuohjelmistoja ei siis esitellä tässä raportissa, mutta niistä löytyy lisätietoa toimeksiantajan JAMES-käsikirjasta, jossa JAMES-menetelmän käyttö on ohjeistettu.

JAMES-menetelmään liittyvää kehitys-, tiedotus-, neuvonta- ja koulutustyötä suoritetaan menetelmätiimiin kuuluvassa JAMES-tuessa, johon kuuluu useita henkilöitä. Menetelmään liittyvä teoretinen tieto sekä kuviot ovat siis syntyneet useiden henkilöiden yhteistyönä pitkän ajan kuluessa. Tässä opinnäytetyössä on käytetty JAMES-menetelmän lähteinä erityisesti JAMES-käsikirjaa sekä 6.5.2011 järjestetyn Kelan sisäisen menetelmäinfotilaisuuden esitysmateriaaleja. Työssä esitelty JAMES-menetelmään liittyvät kuviot pohjautuvat RUP:iin, ja niitä on muokattu erityisesti toimeksiantajan käyttöön soveltuviksi. Lisäksi on huomioitava, että JAMES-menetelmää kehitetään jatkuvasti menetelmätiimin toimesta. Opinnäytetyöprosessin alkaessa menetelmästä oli käytössä versio 3.1, ja lähiaikoina julkaistaan versio 4.0.

Toimeksiantajaorganisaatiossa JAMES-menetelmää hyödyntäviä ohjelmistotuotanto-projekteja kutsutaan yleensä JAMES-projekteiksi erona perinteistä vesiputousmallia hyödyntäviin projekteihin. JAMES-projektit pohjautuvat ohjelmistoprojektin kehitystyön jakamiseen osiin, joita ovat elinkaarivaiheet, iteraatiot, työvaiheet, tehtävät, roolit, lähtöaineistot ja lopputulokset. Niiden kautta määrittyy, mitä tehdään projektin missäkin vaiheessa ja kenen toimesta. (Savolainen, 2011.)

4.1 JAMES-menetelmän rakenneosat

Elinkaarivaiheet (Kuvio 4.) ovat JAMES-projektin ydin; ne vaiheistavat projektin ja painottavat projektin tehtäviä. Elinkaarivaiheita ovat Aloitus, Tarkennus, Rakennus ja Siirtyminen. Nämä vaiheet seuraavat toisiaan ajallisessa järjestyksessä, ja niillä jokaisella on oma tavoitteensa ja merkityksensä. Aloitusvaiheessa varmistetaan, että kaikilla projektiin osallistuvilla henkilöillä sekä projektin sidosryhmillä on yhteinen näkemys siitä, mitä ollaan tekemässä. Vaiheessa kiinnitetään yhteiset tavoitteet ja aletaan hahmotella järjes-

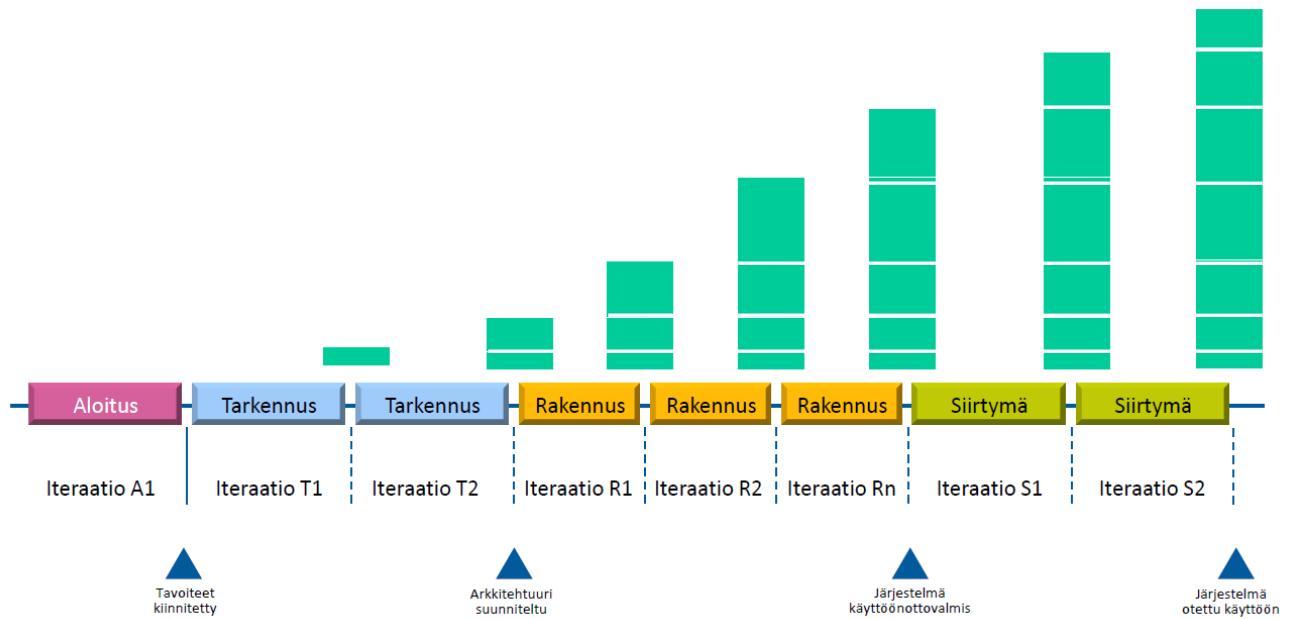
telmän toiminnallisuutta. Tarkennusvaiheessa tarkennetaan vaatimuksia ja ratkaistaan mahdollisia ongelmakohtia. Rakennusvaiheessa tuotetaan kohdejärjestelmää ja testataan sen toimivuutta. Siirtymävaiheessa otetaan valmis järjestelmä käyttöön ja huolehditaan loppukäyttäjien koulutuksesta. (Savolainen 2011; Turtia 2011.)

Kunkin elinkaarivaiheen lopussa on tarkistuspiste, jossa tutkitaan, onko kyseisen vaiheen tehtävät tehty ja tavoite täytetty. Jos tavoitteet on täytetty, voidaan siirtyä seuraavaan elinkaarivaiheeseen. Muussa tapauksessa jatketaan nykyisen vaiheen tehtävien tekemistä niin kauan, että tavoitteet on saavutettu. Elinkaarivaiheet eivät siis ole ajallisesti rajattuja, vaan vaiheet jatkuvat niin kauan, että niihin sisältyvät tavoitteet on saavutettu, tai kunnes projektin aikataulu pakottaa siirtymään eteenpäin. Tällaisessa tapauksessa kesken jääneet tehtävät siirretään suoritettavaksi seuraavan vaiheen alussa. (Savolainen 2011; Turtia 2011.)



Kuvio 4. JAMES-projektin elinkaarivaiheet ja kolmioilla merkityt esimerkit tarkistuspisteistä (Turtia 2011)

Iteraatiot ovat toinen JAMES-menetelmään liittyvä rakenneos. Iteratiivisessa ohjelmistokehitysprosessissa suunnittelua ja toteutusta tehdään pienissä toistuvissa osissa eli iteraatioissa. Iteraatiot seuraavat toisiaan ikään kuin spiraalimaisesti siten, että edellisen kierroksen lopputulos on seuraavan kierroksen lähtökohta. JAMES-menetelmän iteraatioihin liittyy ajatus inkrementaalisuudesta. Inkrementaalisuudella tarkoitetaan sitä, että lopputulosta valmistellaan pienissä osissa lisäyksittäin sen sijaan, että pyritäisiin tekemään valmis lopputulos yhdellä kerralla. Kussakin iteraatiossa on tavoitteena tehdä tietty osakokonaisuus eli inkrementti lopullisesta ohjelmasta. Iteraatioiden ja inkrementaalisuuden suhdetta toisiinsa esitellään kuviossa 5, jossa elinkaarivaiheiden iteraatioiden myötä lisääntyvät vihreät palaset kuvaavat yksittäisiä inkrementtejä eli kehitettävän järjestelmän osia. (Savolainen, 2011.)



Kuvio 5. Iteratiivisuus ja inkrementaalisuus JAMES-projekteissa (Turtia 2011)

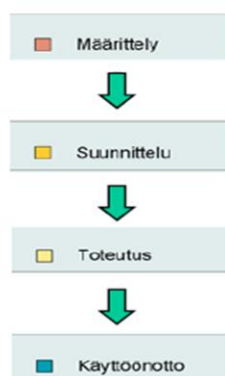
Iteraatioita kuvaavat kuvion 5 alareunassa olevat iteraatiomerkinnot. A1 on Aloitusvaiheen iteraatio, ja T1 ja T2 ovat Tarkennusvaiheen iteraatioita. R1, R2... Rn kuvaavat Rakennusvaiheen iteraatioita, ja lopuksi S1 ja S2 kuvaavat Siirtymävaiheen iteraatioita. Iteraatioiden lukumäärä kunkin elinkaarivaiheen sisällä riippuu projektista ja sen laajuudesta. Iteraatioiden kesto projektissa on yleensä vakio, esimerkiksi 4, 6 tai 8 viikkoa. (Turtia 2011.)

JAMES-projekteissa laaditaan erilaisia iteraatiosuunnitelmia. Projektin alussa laaditaan yleensä kokonaissuunnitelma, joka on karkean tason suunnitelma. Siinä hahmotellaan elinkaarivaiheiden sisältämien iteraatioiden lukumäärät ja niiden sisällöt yleistasolla. Kokonaissuunnitelma päivittyy projektin edetessä. Lisäksi jokaiselle yksittäiselle iteraatiolle laaditaan oma suunnitelma, jossa kuvataan kyseisen iteraation tavoitteet ja tehtävät tarkalla tasolla. Ideaalitapauksessa iteraation tavoitteeksi valittu toiminnallisuus saadaan määriteltä, suunniteltua, toteutettua ja testattua iteraation aikana. Iteratiiviselle toimintatavalle on siis tyypillistä se, että iteraatiosuunnitelmat ovat aluksi melko epämääräisiä ja tarkentuvat projektin edetessä. (Turtia 2011.)

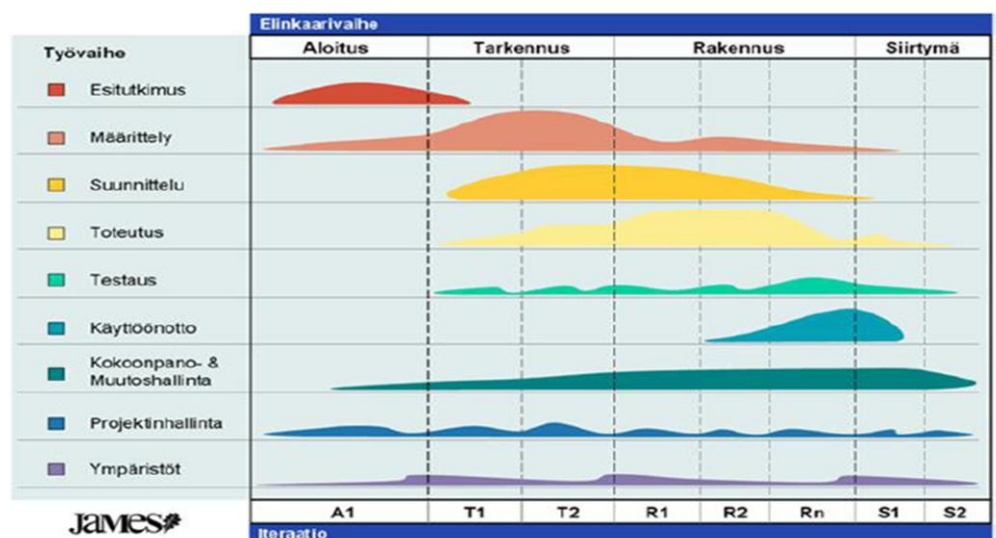
Työvaiheet ovat kolmas JAMES-menetelmän rakenneosia. JAMES-projektin työvaiheita ovat esitutkimus, määrittely, suunnittelu, toteutus, testaus, käyttöönotto, kokoonpano- ja muutoshallinta, projektinhallinta sekä ympäristöt. JAMES-menetelmän työvaiheet kuvastavat vesiputousmallin vaiheita. Työvaiheet liittyvät elinkaarivaiheisiin siten, että jokaisessa elinkaarivaiheessa tehdään samanaikaisesti useita työvaiheita. Kuviossa 6 esitetään JAMES-menetelmän kokonaisajatus sekä vesiputousmallin ja JAMES-menetelmän välinen periaatteellinen ero. Kun vesiputousmallissa työvaiheet seuraavat toisiaan kronologisesti peräkkäin, JAMES-mallissa työvaiheet ovat iteratiivisia ja toistuvat useaan kertaan projektin aikana. (Savolainen 2011.)

Iteratiivisuutta kuvastavat kuviossa 6 esiintyvät eriväriset aallot, jotka osoittavat, mitä työvaihetta on tarkoitus tehdä missäkin elinkaarivaiheessa. Mitä paksumpi aalto on, sitä aktiivisemmin sen työvaiheen tehtäviä tehdään kyseisen elinkaarivaiheen aikana. Kuviossa 6 voidaan myös havaita, että työvaiheet ulottuvat yleensä monen elinkaarivaiheen ylitse, ja niiden sisältämiä tehtäviä tehdään projektin aikana erilaisilla painotuksilla. Esimerkiksi määrittelyä tehdään lähes koko projektin ajan, vaikka se painottuu erityisesti Tarkennusvaiheessa. (Savolainen 2011.)

Vesiputousmallin vaiheet



JAMES-mallin työvaiheet suhteessa elinkaarivaiheisiin



Kuvio 6. Vesiputousmallin ja JAMES-systeemityömenetelmän vertailu (Savolainen 2011)

4.2 JAMES-menetelmään liittyvät tehtävät

Edellä esitellyt elinkaarivaiheet, iteraatiot ja työvaiheet ovat JAMES-menetelmän rakenneseosia, jotka jaksottavat työn etenemistä. Projektin varsinainen työ tehdään tehtävissä, joita suorittamalla projekti etenee suunnitelmien mukaisesti. Jokaiselle tehtävälle on etukäteen määritetty roolit, lähtöaineistot ja lopputuotteet. JAMES-projektin tehtävät lopputuotteineen ja lähtöaineistoineen muodostavat eräänlaisia polkuja. Esimerkiksi tietyn tehtävän X lopputuote toimii tehtävän Y lähtöaineistona, ja puolestaan tehtävän Y lopputuotteet toimivat tehtävien Z ja ZZ lähtöaineistoina. Hyvin suunnitellut, selkeät siirtymät tehtävästä toiseen eli toimivat polut mahdollistavat sujuvan JAMES-projektin. (Turtia 2011.)

Tehtävät ovat RUP:sta JAMES-menetelmään poimittuja, selkeästi määriteltyjä osakokonaisuuksia. Tyypillisiä tehtäviä ovat esimerkiksi Laadi järjestelmän yleiskuvaus, Koordinoi testaus tai Asenna julkaisu. Tehtävät liittyvät aina laajempiin, elinkaarivaihekohtaisiin tehtäväkokonaisuuksiin. Esimerkiksi JAMES-projektin Aloitusvaihe sisältää viisi tehtäväkokonaisuutta: Tavoitteen kuvaaminen, Järjestelmän hahmottaminen, Alustava käyttötapausmallin laatiminen, Testauksen suunnittelemisen ja Työympäristöjen valmistelu. Näistä kokonaisuuksista esimerkiksi Tavoitteen kuvaaminen sisältää neljä tehtävää, jotka ovat: Laadi vaatimustenhallintasuunnitelma, Kerää sidosryhmien tarpeet, Laadi Visio ja Laadi sanasto. Joitakin tehtäviä tehdään tarkentaen koko JAMES-projektin ajan. Esimerkiksi tehtävä Laadi käyttötapausmalli aloitetaan projektin aloitusvaiheessa, mutta sitä tarkennetaan iteroiden projektin tarkennus- ja rakennusvaiheissa. (Turtia 2011.)

Rooli tarkoittaa toimijaa, joka suorittaa tehtävän. Jokaiseen JAMES-projektin tehtävään on kiinnitetty yksi tai useampia rooleja, jotka suorittavat kyseiset tehtävät. Tehtäville nimetään yleensä ensisijainen suorittajarooli sekä täydentäviä suorittajarooleja. Roolit on jaettu neljään pääryhmään: määrittelyroolit, suunnittelu- ja toteutusroolit, testausroolit sekä hallinnolliset roolit. Roolit eivät ole henkilö-, ammattinimike- tai organisaatiosidonnaisia, vaan ne muodostuvat kunkin tehtävän suorittamiseen vaadittavasta osaamisesta. Jos projektin tehtävänä on esimerkiksi käyttöliittymän suunnittelu, voidaan tehtävää suorittavaksi rooliksi nimetä käyttöliittymäsuunnittelija. Projekteissa

voi olla esimerkiksi seuraavia rooleja: käyttöliittymäsuunnittelija, määrittelijä, tietokantasuunnittelija tai projektipäällikkö. Mikäli joku tietty henkilö täyttää useamman tehtävän osaamisvaatimukset, hän voi toimia useassa eri roolissa. (Turtia 2011.)

Jokaiselle tehtävälle on määritelty yksi tai useampi **lähtöaineisto**. Lähtöaineisto eli input määrittelee sen aineiston, joka on otettava huomioon tehtävää suoritettaessa. Lähtöaineistot voivat olla pakollisia tai valinnaisia. Pakollinen lähtöaineisto on ehdottomasti huomioitava tehtävän suorittamisessa, kun taas valinnaista lähtöaineistoa voidaan hyödyntää tarvittaessa. Yleensä lähtöaineistona toimii jonkin edellisen tehtävän lopputuote tai -tuotteet. Esimerkiksi JAMES-projektin Aloituvaiheen tehtävälle Laadi Käyttötapausmalli(kokoava) on määritelty pakolliseksi lähtöaineistoksi aiemman tehtävän lopputuote Visio-dokumentti. (Turtia 2011.)

Roolien ja lähtöaineistojen lisäksi tehtäville on määritelty tietty **lopputuote** eli output. Lopputuotteella tarkoitetaan sitä tuotosta, joka tehtävän aikana on tarkoitus saada aikaan. Lopputuote voi olla esimerkiksi sanallinen dokumentti, määrittelykaavio, koodi tai aineisto. Esimerkiksi JAMES-projektin Aloituvaiheen Kerää sidosryhmien tarpeet - tehtävän lopputuotteena on Sidosryhmien tarpeet -dokumentti. Joskus lopputuotteet voivat myös koostua muista lopputuotteista; esimerkiksi Kokoava käyttötapausmalli koostuu Käyttötapauskaaviosta sekä Käyttötapauskuvauksesta. Usein lopputuotteita tuotetaan tehtävissä iteratiivisesti projektin aikana, jolloin ne täydentyvät ja tarkentuvat projektin edetessä. (Turtia 2011.)

5 Toimeksiannon toteuttaminen

Toimeksiantajaorganisaatiossa alkoi vuonna 2002 tietoarkkitehtuurin kehittämishanke. Sen aikana todettiin, että organisaatiossa käsiteltävään tietomassaan liittyvän metadatan käsittelyssä ja paikantamisessa on puutteita ja ongelmia. Metadatalle tarkoitetaan tietoa tiedosta eli kuvausta esimerkiksi siitä, mikä on tiedon nimi, tietotyyppi ja tiedon pituus (Hovi, Ylinen & Koistinen 2001, 110–111). Tietoarkkitehtuuria päätettiin kehittää rakentamalla tietohakemisto, joka koostui tietokannasta ja Tietohakemisto-sovelluksesta. Organisaatiossa käytössä olevien tietojen tietokuvaukset eli metadata tallennetaan tietokantaan. Tietohakemisto-sovelluksen avulla käyttäjät voivat selata, lisätä, muokata tai poistaa kantaan tallennettuja tietokuvauksia. Tietohakemistossa kuvaukset ovat kaikkien organisaatiossa työskentelevien käytettävissä yhden käyttöliittymän kautta, ja kuvausten ylläpito on helppoa. Tulevaisuuden tavoitteena on, että tietokantaan tallennettuja kuvaustekstejä voitaisiin hyödyntää ohjelmallisesti toimeksiantajan tilastointiin ja raportointiin liittyvissä tietojärjestelmissä. (Rinta-Paavola 2012.)

Myös toimeksiantajani edustaja on ollut aktiivisesti mukana organisaation tietoarkkitehtuurin kehittämisprosessissa. Hän oli havainnut tiettyjä ongelmia erityisesti tilastointi- ja raportointijärjestelmiin liittyvien tiedostojen, taulujen ja muuttujien seliteteksteissä. Selitetekstit saattoivat olla virheellisiä tai puuttua kokonaan, mikä saattoi johtaa väärinkäsityksiin ja vaikeuttaa toimihenkilöiden työskentelyä. Kun organisaation operatiivisten tietojen, taulujen ja muuttujien tietokuvausten tallennus päätettiin organisoida tietohakemiston avulla, löytyi mahdollisuus ratkaista myös edellä mainittuja ongelmia. Jos tilastointi- ja raportointijärjestelmien tiedostojen, taulujen ja muuttujien selitetekstit voitaisiin tuottaa ohjelmallisesti suoraan tietohakemiston tietokannasta, ne olisivat helposti saatavilla, ajan tasaisia ja valideja. (Rinta-Paavola 2012.)

Tietohakemistoon tallennettuja kuvaustekstejä ei vielä toistaiseksi voida hyödyntää ohjelmallisesti, sillä soveltuvaa menetelmää ei ole. Tekstien ohjelmallinen hyödyntäminen tilastointi- ja raportointijärjestelmissä edellyttää purkuajoista, purkutiedostoista, yhdistelytiedostoista ja makro-ohjelmista koostuvaa tietojärjestelmää. Sen avulla esimerkiksi järjestelmissä olevien muuttujien otsikotiedot voitaisiin tuottaa suoraan tietohakemis-

ton tietokannasta sen sijaan, että ne kirjoitetaan käsin suoraan koodiin. Uusi tietojärjestelmä ja toimintatapa parantaisivat organisaation metadatan laatua, vähentäisivät manuaalista työtä ja helpottaisivat tilastointi- ja raportointijärjestelmien ylläpitoa. (Rinta-Paavola 2012.)

5.1 Toimeksiannon esittely

Opinnäytetyöni toimeksianto liittyi edellä kuvatun tietojärjestelmän kehittämiseen. Toimeksiannon mukaisesti minun tuli toteuttaa Kelan Tietohakemiston tekstejä hyödyntävän tietojärjestelmän kehitystyön Aloitusvaihe ja tuottaa sovitut määrittäydokumentit. Muut vaiheet tullaan suorittamaan opinnäytetyöprosessin päätyttyä. Projektin alkuvaiheessa tehtyjen rajausten mukaisesti opinnäytetyön aikana määriteltävä tietojärjestelmä keskittyy kuvaustekstien hyödyntämiseen erityisesti SAS-teknologialla toteutetuissa tilastointi- ja raportointijärjestelmissä toimeksiantajan Mainframe-ympäristössä. Muita teknologioita ja ympäristöjä palvelevat ratkaisut tullaan mahdollisesti toteuttamaan myöhemmin.

Kehitystyö tuli suorittaa toimeksiantajan JAMES-menetelmää soveltaen, mikä tarkoittaa kehitystyön organisoimista ja dokumentaation tuottamista menetelmässä määritellyllä tavalla. JAMES-menetelmää on aikaisemmin sovellettu lähinnä JAVA-teknologialla toteutettavissa operatiivisten järjestelmien kehitysprojekteissa. Organisaatiossa on ryhdytty perehdyttämään ja kannustamaan henkilöstöä JAMES-menetelmän käyttöön myös muihin teknologioihin perustuvissa tietojärjestelmäprojekteissa. Opinnäytetyöni aikana aloitettava tietojärjestelmä tulee toimimaan pilottina JAMES-menetelmän hyödyntämisessä SAS-teknologialla toteutettavien tilastointi- ja raportointijärjestelmien kehittämisessä. Tarkoituksena oli, että toimeksiantaja voi tulevaisuudessa hyödyntää opinnäytetyössäni tuotettua dokumentaatiota esimerkkinä ja mallipohjina vastaavanlaisille tilastoinnin ja raportoinnin tietojärjestelmäkehitysprojekteille.

Jouduin tekemään opinnäytetyöni aikana tiettyjä rajoituksia, jotka aiheuttivat pieniä eroavaisuuksia tavallisiin JAMES-projektien toimintatapoihin verrattuna. JAMES-projekteihin liittyvät kiinteästi eri roolit, jotka suorittavat projektin tehtäviä. Koska opinnäytetyöni tavoitteena olevan tietojärjestelmän kehittäminen oli hyvin pieni projek-

ti, ja koska yksi tavoitteista oli perehdyttää minut toimeksiantajan järjestelmiin ja käytäntöihin, toimin itse eri rooleissa. Tästä johtuen en ota tässä raportissa kantaa rooleihin sen tarkemmin. Suuremmissa projekteissa, joissa työskentelee useita ihmisiä eri rooleissa, olisi välttämätöntä määritellä ja kuvata eri roolit ja niiden tehtävät hyvin tarkalla tasolla, jotta ei tehtäisi päällekkäistä työtä. Toinen huomioitava asia on, että toimeksiantajan JAMES-projekteissa käytetään usein tiettyjä apuohjelmistoja esimerkiksi vaatimusten määrittelyssä ja kaavioiden mallinnuksessa. Tässä opinnäytetyössä näitä välineitä ei kuitenkaan käytetty, sillä opinnäytetyön aikarajoissa minulla ei ollut mahdollista perehtyä näiden välineiden toimintaan ja käyttöön. Varsinaisissa JAMES-projekteissa niiden käyttö olisi suositeltavaa.

5.2 Kehitysprojektin Aloitusvaiheen toteuttaminen ja dokumentointi

JAMES-projektin Aloitusvaiheen tarkoituksena on varmistaa, että kaikilla projektiin osallistuvilla henkilöillä ja projektin sidosryhmillä on yhteinen näkemys siitä, mitä ollaan tekemässä. Vaiheen tehtäväkokonaisuuksia ovat tavoitteen kuvaaminen, järjestelmän hahmottaminen, alustavan käyttötapausmallin laatiminen, testauksen suunnitteleminen sekä työympäristöjen valmisteleminen. Vaiheen lopussa tarkistuspisteessä varmistetaan, että tavoitteet on kiinnitetty, ennen kuin voidaan siirtyä seuraavaan vaiheeseen. Mikäli tehtäviä ei ole saatu valmiiksi tai vaiheen lopputuotteissa on jotakin korjattavaa, voidaan työtä niiden viimeistelemiseksi jatkaa seuraavassa iteraatiossa. Tämänkin iteraation päätteeksi seuraa tarkistuspiste ennen seuraavaan vaiheeseen siirtymistä. (JAMES-käsikirja 2012.)

Taulukossa 1 on lueteltu kaikki JAMES-projektin Aloitusvaiheen viisi tehtäväkokonaisuutta ja niissä tuotettavat 17 lopputuotetta. Lopputuotteet ovat erilaisia määrittelydokumentteja ja aineistoja, jotka toimivat apuna järjestelmän kehitystyössä. Toimeksiantajan mukaisesti kehitettävä tietojärjestelmä liittyy SAS-teknologialla toteutettavaan datankäsittelyyn, eikä kaikkien 17 lopputuotteen tuottaminen ole mielekästä tämän tyyppisissä projekteissa. Esimerkiksi käyttöliittymän suunnitteluun ja laatimiseen liittyvät dokumentit jätettiin pois, koska kehitystyön kohteena oleva tietojärjestelmä tulee toimimaan Mainframe-ympäristössä eikä siinä ole erillistä käyttöliittymää. Tuotettavat lopputuotteet valittiin yhdessä toimeksiantajan ja JAMES-tuen edustajan kanssa. Tau-

lukon kolmannessa sarakkeessa kerrotaan, päätettiinkö kyseinen lopputuote tuottaa tämän opinnäytetyön aikana.

Taulukko 1. JAMES-projektin Aloitusvaiheen tehtäväkokonaisuudet, niihin liittyvät lopputuotteet sekä opinnäytetyössä tehtyt lopputuotteisiin liittyvät päätökset

Tehtäväkokonaisuus	Lopputuote	Päätös
Tavoitteen kuvaaminen	Vaatimustenhallintasuunnitelma	Ei tuoteta
	A. Sidosryhmien tarpeet	Tuotetaan
	B. Visio	Tuotetaan
	C. Sanasto	Tuotetaan
Järjestelmän hahmottaminen	Prosessikaavio	Ei tuoteta
	D. Järjestelmän yleiskuvaus	Tuotetaan
	Täydentävät vaatimukset	Ei tuoteta
	Käytettävyyssuunnitelma	Ei tuoteta
	E. Tietoturvaselvitys	Tuotetaan
Alustava käyttötapausmallin laatiminen	Käyttöliittymähahmotelma	Ei tuoteta
	Käyttötapausmalli (kokoava)	Ei tuoteta
	- Käyttötapauskaavio	
	- Aktiviteettikaavio	
	- Käyttötapauskuvaus	
	- Sekvenssikaavio	
	- Luokkamalli (looginen)	
Testauksen suunnitteleminen	Testauksen työsuunnitelma	Ei tuoteta
	F. Systeemitestisuunnitelma	Tuotetaan
Työympäristöjen valmistelu	Projektin työympäristöjen tilauskaavake	Ei tuoteta
	Projektin työympäristöt	Ei tuoteta
	Ajoympäristöjen tilauskaavake	Ei tuoteta
	Ajoympäristöt	Ei tuoteta

Aloitusvaiheen lopputuotteista kuuden lopputuotteen arvioitiin olevan tarpeellisia opinnäytetyössäni ja sitä vastaavissa tietojärjestelmäkehitysprojekteissa. Opinnäytetyö-

prosessin aikana päätettiin tuottaa seuraavat kuusi lopputuotetta malliksi tuleville projekteille: Sidosryhmien tarpeet -dokumentti, Visio-dokumentti, Sanasto-dokumentti, Järjestelmän yleiskuvaus -dokumentti, Tietoturvaselvitys-dokumentti sekä Systeemitietosuunnitelma-dokumentti. Tuotetut JAMES-lopputuotteet on sijoitettu opinnäytetyön liitteiksi. Opinnäytetyön yhteydessä julkaistavia lopputuotteita on muokattu siten, että toimeksiantajan kannalta liikesalaisuuksiksi luokiteltavat asiat on muotoiltu yleisemmälle tasolle. Seuraavaksi esittelen tuotettavaksi valitut lopputuotteet, kerron perustelut niiden valinnalle ja kuvailen, miten lopputuotteita työstettiin opinnäytetyöprosessin aikana.

A. Tuotettu lopputuote: Sidosryhmien tarpeet -dokumentti (liite 2)

JAMES-projektien ensimmäinen tuotettava lopputuote on Sidosryhmien tarpeet -dokumentti. Sidosryhmillä tarkoitetaan tässä niitä tahoja, jotka ovat jollakin tavalla mukana valmiin järjestelmän käyttämisessä tai hyödyntämisessä. Jotta kehitettävä järjestelmä voitaisiin rakentaa toimivaksi ja jotta se vastaisi tarpeisiin mahdollisimman hyvin, on etukäteen selvitettävä, mitä sidosryhmät tulevalta järjestelmältä odottavat. Sidosryhmien tarpeet -dokumenttiin kirjataan sidosryhmien vaatimukset, toiveet ja odotukset järjestelmälle. Sidosryhmien tarpeet -lopputuote kuuluu Tavoitteen kuvaaminen -tehtäväkokonaisuuteen, ja se toimii lähtöaineistona useille muille JAMES-lopputuotteille, kuten Visiolle ja Käytettävyyssuunnitelmalle. Dokumenttiin kerättyjen tarpeiden pohjalta generoidaan myös järjestelmälle kohdistetut vaatimukset. Sidosryhmien tarpeet -dokumentti on JAMES-projektien tärkeimpiä lopputuotteita, sillä siinä luodaan pohja projektille ja kehitettävälle järjestelmälle. (JAMES-käsikirja 2012.)

Sidosryhmien tarpeet -dokumentista oli olemassa karkea mallipohja, jonka rakenne osoittautui kuitenkin toimimattomaksi. Pohja oli laadittu lähinnä tueksi haastatteluille, ja useamman sidosryhmän tarpeet oli vaikeaa saada kirjattua siihen selkeästi. Keskusteltuani JAMES-tuen edustajan kanssa muokkasin mallipohjaa toimivammaksi siten, että kukin sidosryhmä esitetään omana kappaleenaan. Alakappaleiksi lisättiin jako kunkin ryhmän ensisijaisiin ja toissijaisiin tarpeisiin.

Toimeksiantajan kanssa käytyjen keskusteluiden perusteella tunnistin kolme pääasiallista sidosryhmää. Ensimmäisessä sidosryhmässä olivat sellaiset tahot, jotka tulevat käyttämään kohdejärjestelmää eli SAS-teknologialla toteutettuja tilastointi- ja raportointijärjestelmiä kehittävät ja ylläpitävät henkilöt. Toinen sidosryhmä muodostui henkilöistä, jotka hyötyvät kehitettävän tietojärjestelmän toiminnallisuudesta: tilastollisen materiaalin loppukäyttäjät. Kolmas sidosryhmä oli SAS-sovellusarkkitehtuurin ja tietoarkkitehtuurin kehittäjät, joilla oli erilaisia arkkitehtuurien kehittämiseen liittyviä odotuksia ja tarpeita kehitettävälle järjestelmälle. Näiden kolmen pääasiallisen sidosryhmän lisäksi tunnistin neljännen sidosryhmän, joka koostui useista erillisistä tahoista. Nämä tahot olivat sellaisia, joiden on tarpeen saada tietoa meneillään olevasta kehitystyöstä ja tulevasta tietojärjestelmästä.

Toimeksiantajani kanssa käydyissä keskusteluissa sain tietooni edellä mainittujen sidosryhmien edustajat, joita olisi tarpeellista haastatella toiveiden ja tarpeiden kartoittamiseksi. Laadin haastattelukysymykset itsenäisesti, minkä jälkeen toimeksiantaja katselmoi ne. Tämän jälkeen tarkensin ja muokkasin joitakin kysymyksiä vielä hieman. Seuraavaksi varasin haastatteluajat ja lähetin haastattelun teemat haastateltaville etukäteen tutustuttavaksi. Haastattelukysymykset löytyvät Sidosryhmien tarpeet -dokumentin liitteistä. Haastatteluissa käsiteltiin muun muassa seuraavia teemoja: sidosryhmän edustajan taustatiedot ja työtehtävät, edustettu käyttäjäryhmä sekä sidosryhmän luotettavuus-, käytettävyys-, suoritus- ja tukitarpeet tai vaatimukset kehitettävälle tietojärjestelmälle.

Tein yhteensä 8 haastattelua. Haastattelut olivat luonteeltaan konsultatiivisia, ja suoritin ne yksilöhaastatteluina tehden tarvittaessa tarkentavia kysymyksiä. Kirjasin haastatteluista muistiinpanoja, ja suoritettuani kaikki haastattelut laadin niiden pohjalta projektin ensimmäisen JAMES-lopputuotteen, Sidosryhmien tarpeet -dokumentin. Dokumentissa luokittelin keräämäni tarpeet kunkin sidosryhmän kohdalla omien näkemysteni perusteella kahteen ryhmään: ensisijaisiin ja toissijaisiin tarpeisiin. Osa tarpeista oli sellaisia, että kaksi tai useampi sidosryhmä esitti saman tarpeen. Esimerkiksi useampi kuin yksi sidosryhmä esitti seuraavat tarpeet: tietojärjestelmän käytön tulisi olla yksinkertaista, tietojärjestelmän kehitystyön on tapahduttava standardien mukaisesti, ja kehitystyö on dokumentoitava hyvin. Ristiriitaisia, toisilleen vastakkaisia tarpeita ei tullut esille.

Kunkin sidosryhmän edustajat katselmoivat kirjaamani tarpeet ja saivat ottaa kantaa tekemääni luokitteluun. Katselmoinnista saamani kommenttien pohjalta tein dokumenttiin vielä pieniä lisätarkennuksia. Lopuksi toimeksiantaja katselmoi ja hyväksyi dokumentin lopullisen version 1.0 projektin lopputuotteeksi.

B. Tuotettu lopputuote: Visio-dokumentti (liite 3)

Sidosryhmien tarpeet -dokumentin sekä muun mahdollisesti käytettävissä olevan taustamateriaalin pohjalta laaditaan seuraava JAMES-lopputuote, Visio-dokumentti. Myös Visio kuuluu Tavoitteen kuvaaminen -tehtäväkokonaisuuteen, ja se on lyhyt kuvaus siitä, mitä projektilla on tarkoitus saavuttaa. Visio-dokumenttiin kuvataan työn perustana oleva ongelma sekä kootaan projektin tavoitteet, osapuolet, rajaukset, järjestelmän tarpeet ja ominaisuudet yleisellä tasolla. Sen avulla varmistetaan, että kaikilla sidosryhmillä ja projektin jäsenillä on sama näkemys siitä, mitä ollaan kehittämässä ja miksi. Visiossa pysytellään yleisellä tasolla vastaten kysymyksiin MITÄ ja MIKSI. Tarkempaan toiminnallisuuden kuvaamiseen, MITEN, keskitytään muissa lopputuotteissa. Visio toimii lähtöaineistona esimerkiksi Käytettävyyssuunnitelma- ja Kokoava käyttötapausmalli -lopputuotteille. (JAMES-käsikirja 2012.)

Visio on yksi tärkeimmistä lopputuotteista, ja se tulisi laatia jokaisessa JAMES-projektissa. Myös Visiosta oli olemassa karkea mallipohja, jota saatoin hyödyntää projektissani. Mallipohja ei kuitenkaan ollut mielestäni looginen tai intuitiivisesti täytettävä, joten kävin jälleen JAMES-tuen edustajan kanssa keskusteluita dokumentin rakenteesta ja oikeasta täyttämisestä. Keskusteluissa kävi ilmi, että JAMES-tuki on dokumenttipohjan ongelmista tietoinen ja että pohjaa ollaan kehittämässä toimivammaksi seuraavaan JAMES-versioon (4.0). Toimeksiantajani kanssa sovittiin tässä vaiheessa, että en muokkaa mallipohjien rakenteita tätä projektia varten, vaan projektin lopputuotteet Sidosryhmän tarpeet -dokumenttia lukuun ottamatta tuotetaan nykyisen JAMES-version 3.1 mukaisina. JAMES-tuki toivoi, että ilmoittaisin heille mahdollisista mallipohjan käyttöön liittyvistä epäkohdista ja ongelmista, jotta he voisivat huomioida ne JAMES-menetelmän kehitystyössä.

Laadin Visio-dokumentin käyttäen hyväksi Sidosryhmien tarpeet -dokumenttia sekä sen taustamateriaalina käytettyjä haastattelumuistiinpanoja. Joihinkin tekstikohtiin pyysin toimeksiantajalta kommentteja ja tarkennuksia, sillä en ollut täysin varma oikeista sanamuodoista. Myös dokumentin kappaleessa 3.6 esitettyjen tarpeiden priorisoinnissa pyysin apua toimeksiantajaltani, jotta prioriteetit saataisiin määriteltyä oikein.

Priorisointi suoritettiin siten, että jokainen katselmointitilaisuuteen saapuneista toimeksiantajan edustajista antoi Visioon kirjatuille 17 sidosryhmätarpeelle prioriteetteja arvoiltaan 1-3. Sovimme prioriteetin 1 tarkoittavan sitä, että kyseinen tarve on arvioijan mielestä ehdottomasti toteutettava kehitystyössä. Prioriteetti 2 tarkoitti sitä, että tarve on pyrittävä huomioimaan kehitystyössä mahdollisuuksien mukaan. Prioriteetti 3 annettiin sellaisille tarpeille, jotka voidaan tarvittaessa siirtää toteutettavaksi vasta myöhemmissä järjestelmäversioissa. Lopuksi laskin erikseen jokaiselle tarpeelle annettujen prioriteettien keskiarvon, josta tuli kyseisen tarpeen lopullinen prioriteetti. Tämän arvon perusteella tarpeet voitiin asettaa tärkeysjärjestykseen ja ottaa kantaa siihen, mitkä järjestelmän ominaisuudet tulisi ensisijaisesti pyrkiä toteuttamaan.

Dokumentin kappale 11 ja siihen sisältyvät alakappaleet jäivät tässä vaiheessa vielä osittain tarkentumatta, sillä en ehtinyt saamaan JAMES-tuelta ohjeistusta oikeasta täyttömuodosta. Dokumenttia tullaan siis täydentämään iteratiivisesti vielä opinnäytetyön päätyttyä. Toimeksiantaja katselmoi ja hyväksyi dokumentin version 0.6 projektissa edelleen täydennettäväksi lopputuotteeksi.

C. Tuotettu lopputuote: Sanasto-dokumentti (liite 4)

Sanasto-dokumentti on kolmas Tavoitteen kuvaaminen -tehtäväkokonaisuuden lopputuote, joka opinnäytetyöprojektissa päätettiin tuottaa. Sanastoon kirjataan projektissa käytettävät keskeiset termit ja lyhenteet selkokielisine selitteineen, jotta projektin terminologia säilyisi yhtenäisenä. Sanasto helpottaa myös eri osastoilta olevien projektiryhmäläisten kommunikaatiota ja ymmärrystä, jos sovellusalue ei ole entuudestaan tuttu. Sanastoa päivitetään koko projektin ajan. (JAMES-käsikirja 2012.)

Sanasto-lopputuotteesta oli olemassa mallipohja, joka osoittautui selkeäksi ja toimivaksi. Mallipohjassa kukin termi määriteltiin kirjattiin omaksi kappaleekseen, joten kaikki Sanaston sisältämät termit olivat nähtävissä sisällysluettelossa. Vaikka opinnäytetyöprojektissani ei ole useita toimijoita eikä siten myöskään paineita yhtenäisen terminologian muodostamiseen, päätettiin sanasto silti laatia malliksi tuleville projekteille. Sanastosta oli hyötyä paitsi järjestelmän kehitystyön dokumentoimisessa ja siitä viestimisessä, myös oman osaamiseni kehittämisessä, kun perehdyin toimeksiantajani toimialaan ja toimintaympäristöön.

Aluksi kokosin Sanasto-dokumenttiin kaikki sellaiset kehitettävään tietojärjestelmään ja kehitystyöhön liittyvät termit, jotka mielestäni oli tarpeellista määritellä. Seuraavaksi pyysin toimeksiantajaa lisäämään dokumentista mahdollisesti puuttuvat termit, jotka hänen mielestään tuli myös määritellä. Pyrin ensin itse löytämään termeille mahdollisimman kuvaavat määritelmät, ja lopuksi toimeksiantaja tarkisti määritelmät ehdottaen joihinkin pieniä korjauksia. Toimeksiantaja katselmoi ja hyväksyi Sanasto-dokumentin version 1.0 projektin lopputuotteeksi. Myös Sanasto-dokumenttia tullaan täydentämään opinnäytetyön päättyttyä, sillä siihen tulee päivittää kehitystyön myöhemmissä vaiheissa mahdollisesti esille tulevat termit.

D. Tuotettu lopputuote: Järjestelmän yleiskuvaus -dokumentti (liite 5)

Järjestelmän yleiskuvaus -dokumentti kuuluu Järjestelmän hahmottaminen -tehtäväkokonaisuuteen. Dokumentti on yhteenveto järjestelmästä, ja sen tarkoituksena on kuvata toteutettavaa järjestelmää yleisellä tasolla. Järjestelmän yleiskuvaus -dokumentti toimii lähtöaineistona esimerkiksi Systemitestisuunnitelma-lopputuotteelle. Dokumenttiin voidaan sisällyttää esimerkiksi järjestelmää koskevat määrittelyperusteet, yleiset toiminnallisuudet ja järjestelmäarkkitehtuuri. Tämä dokumentti toimii pohjana tietojärjestelmän määrittelylle ja suunnittelulle, ja sitä päivitetään koko projektin ajan iteratiivisesti määrittelyiden täsmentyessä. Järjestelmän yleiskuvaus -dokumentti jää projektin päättyttyä tietojärjestelmän pysyväisdokumentaatioksi, sillä se toimii valmiin järjestelmän käsikirjana. (JAMES-käsikirja 2012.)

Tästä lopputuotteesta oli olemassa valmis mallipohja, jota saatoin hyödyntää dokumentoinnissa. Dokumentin sisältö jäi kuitenkin melko niukaksi, sillä moni Järjestelmän yleiskuvaus -dokumenttiin kirjattava asia tulee selkiytymään vasta projektin myöhemmissä vaiheissa. Kokosin dokumenttiin tässä vaiheessa tiedossa olevia asioita, jotka tulevat tarkentumaan myöhemmin projektin edetessä. Toimeksiantajan edustaja katselmoi ja hyväksyi Järjestelmän yleiskuvaus -dokumentin version 0.3 projektissa edelleen täydennettäväksi lopputuotteeksi.

E. Tuotettu lopputuote: Tietoturvaselvitys-dokumentti (liite 6)

Myös Tietoturvaselvitys-dokumentti kuuluu Järjestelmän hahmottaminen -tehtäväkokonaisuuteen. Dokumentti toimii lähtöaineistona esimerkiksi Ajoympäristöt-lopputuotteelle. Tähän dokumenttiin kuvataan kehitettävälle järjestelmälle asetettavat tietoturvavaatimukset. Samalla siihen kirjataan tietoturvan varmistamiseksi tarvittavat toimenpiteet järjestelmän eri kehittämisvaiheissa. Dokumenttia päivitetään koko projektin ajan. (JAMES-käsikirja 2012.)

Tietoturvaselvitys-lopputuotteesta oli olemassa mallipohja, jota käytin dokumentin pohjana. Mallipohja oli varsin kattava, sillä se oli suunniteltu laajempien, esimerkiksi Kelan ulkopuolisille käyttäjille suunnattujen tietojärjestelmien tietoturvan määrittelemiseen. Kävin toimeksiantajan kanssa useita keskusteluita siitä, mitkä kohdat dokumentissa koskevat tätä projektia ja miten ne tulisi täydentää. Täydensin dokumenttiin tarpeelliset kohdat, mutta jätin myös muut kohdat näkyviin maininnalla: Ei koske tätä projektia. Näin tuottamaani dokumenttia voidaan hyödyntää tulevilla projekteilla ilman, että jokin olennainen asia jäisi määrittelemättä puuttuvan otsikon takia. Lopuksi toimeksiantaja katselmoi ja hyväksyi Tietoturvaselvitys-dokumentin version 1.0 projektin lopputuotteeksi. Myös tätä dokumenttia tullaan tarpeen mukaan päivittämään projektin myöhemmissä vaiheissa, jos esille nousee uusia tietoturvaan liittyviä asioita.

F. Tuotettu lopputuote: Systemitestisuunnitelma-dokumentti (liite 7)

Systemitestisuunnitelma liittyy Testauksen suunnittelu -tehtäväkokonaisuuteen. Vaikka varsinainen testaaminen voidaan aloittaa vasta myöhemmin, aloitetaan testaamisen suunnittelu jo varhaisessa vaiheessa. Systemitestisuunnitelma-dokumentti toimii

projektin testaussuunnitelmana, ja siinä kuvataan systeemitestin tavoitteet, tehtävät, aikataulu, vastuut ja lopputuotteet. Tätä dokumenttia päivitetään projektin edetessä, vähintään jokaisen iteraation suunnittelun yhteydessä. (JAMES-käsikirja 2012.)

Tästä lopputuotteesta ei ollut olemassa valmista mallipohjaa. Laadin Systeemitestisuunnitelma-dokumentin itse hyödyntäen muita JAMES-lopputuotteita sekä Laadi projektin testaussuunnitelma -tehtävän kuvausta. Sain myös JAMES-tuelta nähtäväkseni eräässä toisessa projektissa tuotetun systeemitestisuunnitelma-dokumentin, josta sain hieman apua dokumentin laatimisessa. Suurin osa tähän lopputuotteeseen liittyvästä työstä kuului soveltuvan dokumenttipohjan kokoamisessa, joten sisällöllisiin asioihin tullaan panostamaan vasta myöhemmissä iteraatioissa. Toimeksiantaja katselmoi ja hyväksyi Systeemitestisuunnitelma-dokumentin version 0.2 projektissa edelleen täydennettäväksi lopputuotteeksi.

6 Johtopäätökset ja pohdinta

Tässä opinnäytetyössä perehdyttiin tietojärjestelmäkehitykseen systeemityön vaihejaon ja vaihejakomallien näkökulmasta. Työn teoreettisen osion tavoitteena oli selvittää, mitä vaiheita tietojärjestelmien kehitysprosessiin kuuluu, ja miten ne liittyvät toisiinsa. Työn toiminnallisen osuuden tavoitteena oli aloittaa tietohakemiston tekstejä hyödyntävän tietojärjestelmän kehitystyö ja tuottaa siitä tietyt JAMES-menetelmän mukaiset määrittelydokumentit toimeksiantajan käyttöön. Kolmantena tavoitteena oli perehdyttää opinnäytetyön tekijä toimeksiantajan järjestelmiin ja käytäntöihin tietojärjestelmien kehityksen osalta.

6.1 Yhteenveto

Teoriaosuuden taustakirjallisuuteen tutustuessani havaitsin, että kirjallisuudesta löytyi erilaisia näkemyksiä siitä, kuinka tietojärjestelmäkehitystyötä voidaan vaiheistaa. Jotta opinnäytetyön teoriaosuus ei paisuisi kovin suureksi, päädyin valitsemaan teoriapohjaksi systeemityökurssilta tutun Risto Pohjosen teoksen Tietojärjestelmien kehittäminen. Pohjosen näkemys vaihejaosta ja yleisimmistä vaihejakomalleista tuntui selkeältä, ja arvelin sen tukevan myös toimeksiantajan JAMES-menetelmän esittelyä. Lisäksi oli mielenkiintoista tutkia kurssilla teoriatasolla opeteltuja asioita syvällisemmin ja päästä soveltamaan niitä käytännössä.

Kuten työn teoriaosuudessa kerroin, Pohjosen mukaan tietojärjestelmäkehityksen vaihejako sisältää kahdeksan vaihetta. Vaiheet ovat: esitutkimus-, vaatimusmäärittely-, järjestelmäanalyysi-, suunnittelu-, toteutus-, testaus-, käyttöönotto- ja ylläpitovaihe. Jokaisessa vaiheessa suoritetaan tietyt kyseiseen vaiheeseen liittyvät tehtävät, jotka vievät kehitystyötä eteenpäin. Lisäksi vaiheiden aikana tuotetaan tietyt dokumentit, joihin kirjataan esimerkiksi kohdejärjestelmään liittyviä määrittelyjä ja kuvataan kehitystyötä.

Aluksi tällainen monivaiheinen työskentelytapa kuulosti monimutkaiselta ja aikaa vievältä. Työskentely edellyttää jokaisen vaiheen huolellista läpikäymistä ja suuren dokumenttimäärän tuottamista. Eikö sen sijaan olisi tärkeämpää ryhtyä mahdollisimman pian toteuttamaan kohdejärjestelmää, jotta se valmistuisi nopeasti tuotantoon? Lä-

hemmin asiaan tutustuttuani oivalsin, että systeemityössä määritelty systemaattinen toimintatapa on juuri oikea tapa tehdä tietojärjestelmäkehitystyötä. Vaiheisiin liittyvät tehtävät ja dokumentointi on suoritettava huolella, sillä ne toimivat seuraavien vaiheiden syöteinä ja lähtömateriaaleina. Dokumentaation virheistä tai puutteista seuraa yleensä ongelmia kehitystyön myöhemmissä vaiheissa. Virheiden korjaukset aiheuttavat lisätyötä ja -kustannuksia, ja ongelmat saattavat pahimmillaan johtaa koko kehitystyön epäonnistumiseen. Lisäksi on muistettava, että kehitystyön huolellinen dokumentointi helpottaa tulevaa ylläpitotyötä suuresti.

Teoriaosuudessa totesin, että tietojärjestelmäkehitystyön vaiheiden keskinäinen järjestys ja ajallinen kesto voivat vaihdella eri organisaatioissa tai kehitysprojekteissa. Vaiheiden järjestystä ja ajoitusta on pyritty mallintamaan erilaisten vaihejakomallien avulla. On muistettava, että vaihejakomallit ovat suuntaa-antavia kuvauksia kehitystyön etenemisestä ideaalitapauksessa. Malleja on mahdotonta asettaa paremmuusjärjestykseen, sillä jokin malli saattaa toimia hyvin tietyssä projektissa tai organisaatiossa, kun taas joku toinen projekti tai organisaatio vaatii täysin erilaista lähestymistapaa.

Organisaatioissa on usein tehty linjauksia siitä, millaista vaihejakomallia siellä toteutettavissa tietojärjestelmäprojekteissa noudatetaan. Joissakin organisaatioissa on jopa kehitetty olemassa olevien mallien pohjalta oma vaihejakomalli, joka palvelee juuri kyseisen organisaation tarpeita. Tämän opinnäytetyön toimeksiantajaorganisaation tietojärjestelmäprojekteissa oli aiemmin käytetty vesiputousmallia. JAVA-teknologian yleistymisen oli kuitenkin asettanut uudenlaisia tarpeita systeemityömenetelmälle, ja vesiputousmallin ohelle oli kehitetty erityisesti JAVA-projekteille suunnattu JAMES-systeemityömenetelmä.

JAMES-menetelmän perustana ovat ajatukset iteratiivisuudesta ja inkrementaalisuudesta, eli kohdejärjestelmää rakennetaan sykleissä osa kerrallaan toistaen tietyt vaiheet. JAMES-menetelmässä on siis elementtejä teoriaosuudessa esitellystä spiraalimallista, jossa kehitystyö etenee iteratiivisesti kierros kierrokselta. Lisäksi menetelmässä on havaittavissa myös vesiputousmallin piirteitä. Vesiputousmallin vaiheet ovat JAMES-

menetelmän työvaiheita, ja niitä toteutetaan iteraatioiden sisällä siten, että jokaisessa iteraatiossa suoritetaan ikään kuin pieni vesiputous.

Toimeksiantajan mukaan perinteisillä teknologioilla (esimerkiksi SAS ja PL/1) toteutetuissa projekteissa on käytetty lähinnä vesiputousmallia. Esimerkiksi SAS-teknologialla toteutetut tilastointiin ja raportointiin liittyvät tietojärjestelmät ovat painottuneet datan käsittelyyn, eivätkä ne sisällä varsinaisia käyttöliittymiä. Tällaisten järjestelmien kehitystyölle on tyypillistä, että määrittely tarkentuu työskentelyn edetessä. SAS-teknologialla toteuttavien tietojärjestelmien kehitystyössä vesiputousmallin noudattaminen onkin ollut melko vapaamuotoista. Tuotettu dokumentaatio on saattanut olla hyvin suppeaa, ja sen laatu on riippunut toteuttajastaan. Pahimmassa tapauksessa kaikki tieto kehitetystä järjestelmästä on ollut vain kehittäjänsä muistissa, jolloin ylläpitovastuiden siirto muille henkilöille on ollut monimutkaista vaaten paljon aikaa, selvityksiä ja lisätyötä.

On selvää, että toimeksiantaja hyötyisi paremmin systematisoidusta ja standardoidusta toimintatavasta. Toiminnan kehittämiseksi toimeksiantajani pyysi minua perehtymään JAMES-menetelmään käyttäen mallitapauksena tiettyä SAS-teknologialla toteutettavaa tietojärjestelmäkehitysprojektia. Minun tuli tutkia, kuinka standardoitua JAMES-menetelmää voitaisiin ryhtyä soveltamaan myös SAS-teknologialla toteutetuissa tilastoinnin ja raportoinnin järjestelmien kehitysprojekteissa. Tähän perustuen asetimme toimeksiantajani kanssa opinnäytetyön toiminnallisen osuuden tavoitteeksi kohdejärjestelmän Aloitusvaiheen suorittamisen JAMES-menetelmän mukaisesti. Sen aikana aloitaisin uuden tietojärjestelmän kehitystyön ja tuottaisin dokumentaationa tietyt JAMES-lopputuotteet toimeksiantajan käyttöön.

Eräs opinnäytetyöhön liittyvistä tehtävistäni oli käydä läpi erilaiset Aloitusvaiheen JAMES-lopputuotteet ja valita niistä juuri tämän tyyppisille tietojärjestelmäprojekteille keskeiset lopputuotteet. Havaitsin, että JAMES-menetelmän Aloitusvaiheelle on määriteltä 17 mahdollista lopputuotetta. Tulin siihen johtopäätökseen, ettei niiden kaikkien tuottaminen ollut mielekästä tällaisessa melko pienimuotoisessa datankäsittelyyn painottuvassa kehitysprojektissa. Toimeksiantajan ja JAMES-tuen edustajien kanssa käytöksen keskusteluiden perusteella päädyimme siihen, että tuottaisin opinnäytetyön aikana

kuusi lopputuotetta. Ne toimivat kohdejärjestelmän määrittelydokumentteina sekä mallipohjina vastaavanlaisille datankäsittelyyn liittyville tietojärjestelmäprojekteille. Laati-
miani dokumentteja annettiinkin malleiksi toisille projekteille jo opinnäytetyöprosessin aikana.

Osaan lopputuotteista oli JAMES-tuen toimesta laadittu karkeat mallipohjat, jotka helpottivat lopputuotteen tuottamista. Mallipohjiin oli kirjattu olennaisiksi arvioidut otsikot ja selitykset, mitä kunkin otsikon alle oli ajateltu kirjattavan. Osa valmiista mallipohjista oli kuitenkin melko monimutkaisia tai jopa datan käsittelyyn painottuvien tietojärjestelmien kehitysprojekteihin soveltumattomia, enkä voinut hyödyntää niitä sellaisenaan työskentelyssäni. Sovimme, että muokkaan Sidosryhmien tarpeet - lopputuotteen mallipohjan toimivammaksi, mutta päädyimme pitämään muut lopputuotteet vielä toistaiseksi JAMES 3.1 version mukaisina. Ongelmia prosessin aikana tuotti myös se, että osa lopputuotemalleista oli olemassa vain nimitasolla. Tällöin jouduin itse pohtimaan ja selvittämään, mitä asioita kyseiseen lopputuotteeseen olisi hyvä kirjata.

Lopputuotteiden laatimisvaiheessa kävin useita keskusteluita JAMES-tuen kanssa, ja he katselmoivat ja kommentoivat tuotettujen lopputuotteiden rakennetta. Samalla he saivat tietoa siitä, kuinka kehitetty JAMES-menetelmä soveltuu datan tilastolliseen käsittelyyn painottuviin projekteihin, ja millaisia kehitystarpeita uuteen teknologiaan soveltaminen tuo menetelmälle. Rakennekatselmoinnin lisäksi työskentelyssä suoritettiin säännöllisesti myös tekstisisältöjen katselmointia toimeksiantajan ja tulevien käyttäjäryhmien edustajien kanssa. Tällainen katselmointikäytäntö paransi tuotettavien dokumenttien laatua, kun tuotettavia tekstejä tarkasteli ja kommentoi useampi asiaan hyvin perehtynyt henkilö. Samalla se kuitenkin hidasti opinnäytetyöprosessin etenemistä, sillä yhteisten katselmointiaikojen löytäminen kalenterista ei ollut aivan helppoa. Tulevissa projekteissa on mielestäni tarpeellista suorittaa jokaisen lopputuotteen kohdalla vähintään lopputuotekatselmointi, jotta mahdolliset virheet tai puutteet saadaan oikaistua ennen dokumenttien luovuttamista projektissa eteenpäin. Vaikka tällainen käytäntö viekin jonkin verran aikaa, se voi ajan mittaan tuoda suuria säästöjä eliminoimalla määrittelystä johtuvia virheitä.

Opinnäytetyöprosessin aikana tuottamani lopputuotteet osoittautuivat tärkeiksi ja olennaisiksi dokumenteiksi uuden tietojärjestelmän kehitystyössä. Mielestäni toimik-siantaja voisi harkita niiden tuottamista sellaisissa tilastointi- ja raportointijärjestelmien kehitysprojekteissa, joiden tuloksena on pysyvästi tuotannossa toimiva järjestelmä. Do-kumentoinnin laajuutta ja yksityiskohtaisuutta toki voi aina miettiä meneillään olevan projektin kannalta. Aivan pienessä projektissa ei ole välttämättä mielekästä käyttää suurta määrää työtunteja JAMES-menetelmän mukaisten lopputuotteiden tuottami-seen, vaan esimerkiksi pelkkä Järjestelmän yleiskuvaus -dokumentti voi olla täysin riit-tävä dokumentaatio. Osa dokumenteista ehti jalostumaan sisällöllisesti varsin pitkälle opinnäytetyöprosessin aikana, osa taas jäi melko keskeneräisiksi, ja niitä tullaan täyden-tämään tulevien iteraatioiden aikana. Seuraavaksi esittelen vielä muutamia näkökohtia tuotetuista lopputuotteista.

Sidosryhmien tarpeet -dokumenttiin kirjattiin sidosryhmien järjestelmälle asettamat toiveet ja vaatimukset. Sidosryhmien tarpeiden kartoittaminen haastatteluiden avulla tuntui mielekkäältä toimintatavalta. Haastattelut mahdollistivat erilaisten näkemysten esille tuomisen ja auttoivat minua ymmärtämään sidosryhmien tarpeita ja toimintaedel-lytyksiä. Arvelen, että varaamalla kalenterista ajan kasvotusten tapahtuvaa haastattelua varten sain myös sidosryhmien edustajien huomion asialle tehokkaammin kuin esimer-kiksi sähköpostitse lähetetyn kyselyn avulla. Ennen haastatteluiden tekemistä on kui-tenkin mietittävä tarkkaan, mitkä ovat kyseisen projektin sidosryhmiä, jotta yksikään sidosryhmä ei jäisi vahingossa huomioimatta.

Jälkeenpäin arvioin, että Sidosryhmien tarpeet -lopputuotteeseen liittyen olisi voinut olla hyvä järjestää sidosryhmien edustajille haastatteluiden ja dokumentin koostamisen jälkeen tarpeiden yhteiskatselmointi tai työpaja. Siinä kerätyt tarpeet oltaisiin voitu käsi-tellä yhdessä, jolloin tilaisuudessa esitetyt ajatukset ja kommentit olisivat saattaneet he-rättää hyvää keskustelua ja tuottaa lisää näkökulmia asiaan. Tämä voisi olla hyvä lähes-tymistapa erityisesti silloin, jos sidosryhmiltä on tullut ristiriitaisia tarpeita, joita täytyisi sovittaa yhteen. Opinnäytetyön puitteissa ei tällaisen yhteistilaisuuden järjestämiseen ollut mahdollisuutta, sillä yhteisen ajankohdan löytäminen olisi siirtänyt tarpeiden jatko-työstämistä viikoilla eteenpäin. Lisäksi tarpeiden luokittelu ensisijaisiin ja toissijaisiin

tarpeisiin olisi ollut hyvä suorittaa yhdessä sidosryhmän edustajien kanssa sen sijaan, että tein luokittelun yksin ja vain hyväksyin sen sidosryhmien edustajilla.

Toinen opinnäytetyön aikana tuotettu lopputuote oli Visio-dokumentti. Visiossa luodaan nimensä mukaisesti projektin toimijoille yhteinen käsitys siitä, mitä kehitysprojektissa on tarkoitus tehdä. Lopputuotteen kuvauksessa mainitaan, että Vision olisi hyvä olla tiivis ja lyhyt. Mallipohja on kuitenkin laadittu niin laajaksi ja yksityiskohtaiseksi, että dokumentti paisui 30-sivuiseksi, vaikka yritin ilmaista asiat tiiviisti. Näin pitkiä määrittelydokumentteja ei aina jakseta edes lukea läpi saati sisäistää niiden sisältämiä asioita. Mallipohjan rakennetta tulisikin mielestäni muokata ja tiivistää, jotta se ei paisuisi liian laajaksi. Ehdottaisinkin, että yksityiskohtaisemmat järjestelmän ominaisuudet siirrettäisiin kirjattavaksi johonkin toiseen dokumenttiin, jotta Visio säilyisi lyhyenä ja ytimekkäänä.

Visio-dokumentin kappaleeseen 3.6 liittyvien sidosryhmätarpeiden priorisointi osoittautui mielenkiintoiseksi ja toimeksiantajan kannalta hyväksi toimintatavaksi. Yleensä tietojärjestelmän kehittäjä on päättänyt yksin tai parin henkilön kesken, mitä ominaisuuksia järjestelmään tulee ensisijaisesti sisällyttää. Nyt kokeilimme tarpeiden priorisointia katselmointitilaisuudessa, johon osallistui eri sidosryhmien edustajia. Menettely pakotti kaikki osallistujat ottamaan kantaa tulevan järjestelmän ominaisuuksiin ja jäseni, mitkä ominaisuudet tullaan toteuttamaan nyt ja mitkä siirretään toteutettavaksi seuraavissa järjestelmäversioissa. Lisäksi lopullisten prioriteettien kirjaaminen Visio-dokumenttiin auttaa palauttamaan mieleen järjestelmän jatkokehitysvaiheessa, millaisia ominaisuuksia järjestelmän uuteen versioon tulisi toteuttaa.

Kolmas tuotettu lopputuote, Sanasto-dokumentti, ei ollut tämän projektin kannalta kovinkaan olennainen dokumentti. Tässä projektissa ei ollut mukana useita osapuolia, joiden kesken täytyisi varmistaa yhteinen termistö. Laajemmissa ja varsinkin osastorajat ylittävissä projekteissa sanaston laatiminen sen sijaan on erittäin tärkeää, jotta projektissa käytettävät termit ja niiden määrittelyt saadaan kaikkien tietoon ja vältetään terminologisilta väärinkäsityksiltä.

Neljänneksi tuotettiin Järjestelmän yleiskuvaus -dokumentti. Se toimii järjestelmän kuvausdokumentaationa, ja sitä on tarkoitus päivittää tietojärjestelmän koko elinkaaren ajan. Tällaista dokumenttia voidaan hyödyntää esimerkiksi uuden vastuuhenkilön perehdyttämisessä järjestelmään sekä järjestelmän ylläpitovaiheessa yksittäisten työtehtävien taustamateriaalina. Lisäksi jatkokehitystyötä on helpompi tehdä, kun järjestelmän nykyisestä tilasta ja toiminnasta on olemassa ajan tasalla oleva dokumentaatio.

Viides tuotettu lopputuote oli Tietoturvaselvitys-dokumentti. Siihen kirjataan kehitettävälle järjestelmälle asetettavat tietoturva vaatimukset ja niihin liittyvät toimenpiteet. Tässä projektissa tietoturvaan liittyvät kysymykset eivät ole niin kriittisiä kuin joissakin muissa projekteissa, sillä kehitettävä tietojärjestelmä ei sisällä arkaluontoisia tietoja. Lisäksi se, että järjestelmä tulee vain Kelan IT-osaston käyttöön ja sen käyttäjillä on jo entuudestaan tietyt käyttövaltuudet ja oikeudet toimia Kelan tietojärjestelmissä, teki dokumentista tämän projektin kannalta vähemmän kriittisen. Kehitettävä tietojärjestelmä tulee toimimaan Kelan Mainframe-ympäristössä, joka huolehtii automaattisesti muun muassa käyttöoikeuksista, lokituksesta sekä versioinnista. Nämä asiat voitiin siis suoraan kirjata järjestettävän standardiohjelmiston määritysten mukaisesti. Tietoturvaselvitys-lopputuote päätettiin kuitenkin tuottaa mallipohjaksi sen vuoksi, että jokaisessa – erityisesti datan käsittelyyn liittyvässä – tietojärjestelmäprojektissa on hyvä ottaa huomioon tietoturvaan liittyvät seikat ja dokumentoida ne huolellisesti.

Kuudes ja viimeinen opinnäytetyön aikana tuotettu lopputuote oli Systeemitestisuunnitelma. Kaikkia ohjelmia ja järjestelmiä on tarpeen testata, ennen kuin ne otetaan käyttöön tuotantoympäristössä. Tämä koskee myös opinnäytetyöni kohteena olevaa tietojärjestelmää. Testauksesta on laadittava tarkat ja systemaattiset suunnitelmat, jotka ohjaavat testausta ja mahdollistavat esimerkiksi henkilöresurssien varaamisen hyvissä ajoin. Opinnäytetyön aikana tuotettu systeemitestisuunnitelma jäi sisällöltään melko keskeneräiseksi, sillä aikaa kului mallipohjan laatimiseen. Lisäksi systeemitestisuunnitelman laatiminen vaatii tarkempaa tietoa tulevista käyttötapauksista, jotta dokumenttiin kirjattavat testustehtävät voidaan suunnitella. Näin ollen käyttötapaukset tulisi mallintaa ja kuvata, ennen kuin systeemitestisuunnitelmaa päästään tarkentamaan.

Opinnäytetyön aikana laatimani kuuden lopputuotteen pohjalta kohdejärjestelmän kehittämistä voidaan jatkaa linjatyönä täydentäen ja hyödyntäen jo laadittua dokumentaatiota. Alustava suunnitelma on, että tulen jatkamaan tietojärjestelmän työstämistä opinnäytetyöprosessin päätyttyä. Järjestelmä on tarkoitus saada valmiiksi ja ottaa käyttöön vuoden 2012 aikana. Ennen seuraavaan JAMES-elinkaarivaiheeseen siirtymistä olisi olla hyvä tuottaa vielä Aloitusvaiheeseen kuuluva Käyttötapausmalli (kokoava) -lopputuote, joka sisältää esimerkiksi aktiviteetti- ja käyttötapauskaaviot. Niiden avulla järjestelmän toiminnallisuutta saadaan vielä mallinnettua tarkemmin, mikä helpottaa Tarkennus-, Rakennus- ja Siirtymävaiheissa suoritettavia tehtäviä. Tämän jälkeen on mahdollista käydä läpi tietojärjestelmän kehitystyön seuraavat elinkaarivaiheet. Niihin liittyvät lopputuotteet tulisi käydä läpi samalla tavalla kuin Aloitusvaiheenkin lopputuotteet ja valita niistä erityisesti tilastointi- ja raportointijärjestelmien kehitysprojektissa tuotettavat dokumentit. Lopuksi koossa olisi ne lopputuotteet, jotka otetaan tavoitteeksi tuottaa vastaavanlaisissa datan käsittelyyn liittyvissä tietojärjestelmäkehitysprojekteissa.

Opinnäytetyöprosessin kolmas päätavoite oli perehdyttää työn tekijä toimeksiantajan järjestelmiin ja käytäntöihin tietojärjestelmäkehityksen osalta. Opinnäytetyöprosessin aikana perehdyin esimerkiksi seuraaviin järjestelmiin, käytäntöihin ja teknologioihin: JAMES-systeemityömenetelmä, Mainframe-ympäristö ja sen operoimisessa käytettävä TSO-tekstieditori, SAS-ohjelmisto ja -komentokieli, tilastointi- ja raportointijärjestelmät ja niiden toiminta sekä tiedonhuoltoprosessi toimeksiantajaorganisaatiossa. Näitä kaikkia elementtejä ja tietoja tulen tarvitsemaan tulevissa työtehtävissäni, ja opinnäytetyö tarjosi hyvän mahdollisuuden perehtyä niihin hallitusti ja suunnitelmallisesti. Lisäksi opinnäytetyöprosessin aikana toimeksiantajan kanssa käydyt keskustelut selkiyttivät organisaation toimintaa ja tavoitteita syventäen ymmärrystäni tulevasta työkentästäni.

6.2 Opinnäytetyöprosessi

Opinnäytetyöprosessini alkoi jo vuoden 2011 alussa. Kävin tuolloin keskusteluita toimeksiantajani kanssa mahdollisen opinnäytetyön aiheista. Vähitellen aiheeksi valikoitui tietojärjestelmien kehittäminen, ja systeemityö vaikutti loogiselta näkökulmalta lähestyä asiaa teoreettisella tasolla. Aluksi toimeksiantajan toiveena oli, että suorittaisin opinnäytetyössäni kolmen eri tietojärjestelmätarpeisiin vastaavan tietojärjestelmän kehittämisen.

prosessin. Tämä osoittautui hyvin pian liian laajaksi kokonaisuudeksi opinnäytetyöprosessin aikarajoihin nähden. Valitsimme opinnäytetyön aikana toteutettavan kehitystyön kohteeksi tietojärjestelmän, joka toimeksiantajan näkökulmasta oli kaikkein kriittisin ja kiireellisin. Kehitettävän tietojärjestelmän oli tarkoitus palvella SAS-teknologialla toteutettuja tilastointi- ja raportointijärjestelmiä sekä niitä ylläpitäviä henkilöitä. Työn edetessä rajasimme opinnäytetyöhön raportoitavan kehitystyön koskemaan vain projektin Aloitusvaihetta, muut vaiheet tulen toteuttamaan myöhemmin opinnäytetyöni ulkopuolella.

Alun perin minun oli tarkoitus saada opinnäytetyö valmiiksi jo joulukuksi 2011. Lopputuotteiden työstäminen ja tuottaminen etenivät kuitenkin melko hitaasti, sillä jokaista lopputuotetta katselmoitiin eri kokoonpanoilla niin rakenteen kuin sisällönkin osalta. Tästä johtuen alkuperäisessä aikataulussa pysyminen olisi vaatinut joko toiminnallisen osuuden rajaamista vielä entisestään esimerkiksi vain kahden tai kolmen lopputuotteen tuottamiseen tai dokumentaation laadusta tinkimistä. Kummatkin näistä valinnoista olisivat pienentäneet toimeksiantajan saamaa hyötyä. Päätin mieluummin ottaa lisää aikaa ja tuottaa suunnitellut dokumentit huolella, jolloin aikataulu siirtyi noin yhdeksällä kuukaudella eteenpäin. Työ valmistui elokuussa 2012.

Opinnäytetyön tekeminen toimeksiantajana omalle työpaikalle osoittautui hyväksi mutta ajankäytön kannalta haasteelliseksi tavaksi työskennellä. Hyvää oli se, että kaikki opinnäytetyöhön liittyvä selvitystyö ja konkreettinen tekeminen liittyivät vahvasti omiin työtehtäviin ja niihin perehtymiseen. Ajankäytön haasteellisuus taas johtui siitä, että muut työtehtävät veivät usein alun perin opinnäytetyölle varaamaani aikaa. Joskus olin esimerkiksi ajatellut, että kävisin työaikana tapaamassa jotakin toimeksiantajaorganisaation henkilöä, jolta saisin lisäselvitystä johonkin lopputuotteeseen kirjattavaan asiaan. Sitten tulikin jokin kiireellinen työtehtävä, jonka jouduin priorisoimaan etusijalle, ja opinnäytetyöhön liittyvä asia jäi selvitettäväksi myöhempänä ajankohtana. Tätä ongelmaa helpottivat jonkin verran työnantajani myöntämät opintovapaapäivät. Otin usein työkonenei mukaani kotiin opintovapaapäivien ajaksi ja laitoin sieltä käsin sähköpostiviestejä toimeksiantajaorganisaation henkilöille. Näin sain hoidettua myös opinnäytetyöhöni liittyviä asioita virka-aikana.

Olin opinnäytetyöprosessin aikana yhteydessä sekä toimeksiantajaani että ohjaavaan opettajaani melko säännöllisesti. Toimeksiantajan kanssa kävin lähinnä työn toiminnalliseen osioon liittyviä keskusteluita. Sain häneltä tarvittaessa ohjausta ja neuvoja esimerkiksi dokumenttien tekstien tuottamiseen tai oikeiden henkilöiden löytämiseen organisaatiossa. Lisäksi hän lopuksi luki ja katselmoi koko opinnäytetyön liitteineen sekä palautteenantoa varten että ottaakseen kantaa tekstin julkaisukelpoisuuteen toimeksiantajaorganisaation kannalta. Ohjaavalta opettajalta sain apua ja tukea esimerkiksi työn rajaamiseen sekä raportin rakenteen katselmointiin. Haluankin lämpimästi kiittää molempia henkilöitä heidän antamastaan tuesta ja ohjauksesta.

Kaiken kaikkiaan opinnäytetyö oli hyvin opettavainen ja antoisa prosessi oman tietotaitoni kehittymisen kannalta. Esimerkiksi koulussa systeemityön ja tietohallinnon teoriakursseilla opitut asiat yhdistyivät käytäntöön, ja moni hieman epäselväksi jäänyt asia selkiytyi mielessäni. Lisäksi opin hahmottamaan ja hallitsemaan omaa ajankäyttöäni entistä tehokkaammin, kun työstin opinnäytetyötä työn, perheen ja harrastusten lomassa. Saamaani oppia voin tulevaisuudessa hyödyntää paitsi työpaikallani, myös muussa elämässäni.

Lähteet

Haikala, I., Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt. 12. uudistettu painos. Talentum. Helsinki.

Haikala, I., Märijärvi, J. 2004. Ohjelmistotuotanto. 10. uudistettu painos. Valikko-sarja. Talentum. Helsinki.

Hovi, A., Ylinen, J., Koistinen, H. 2001. Tietovarastot liiketoiminnan tukena. Asiantuntija-sarja. Talentum. Helsinki.

IBM Rational Unified Process (RUP). 2011. Proven best practices for software and systems delivery and implementation and for effective project management. Verkko-julkaisu. Luettavissa: <http://www-01.ibm.com/software/awdtools/rup/>. Luettu: 5.12.2011.

JAMES-käsikirja. 2012. Kansaneläkelaitoksen IT-osaston menetelmätiimin laatima sähköinen käsikirja organisaation intranetissä. Luettu 25.1.2012.

Kela. 2012a. Kelan yleisesite. Luettavissa: [http://www.kela.fi/in/internet/liite.nsf/alias/Yleisesite/\\$File/Kela_yleisesite_fi_net.pdf?OpenElement](http://www.kela.fi/in/internet/liite.nsf/alias/Yleisesite/$File/Kela_yleisesite_fi_net.pdf?OpenElement). Luettu: 24.6.2012.

Kela. 2012b. Kelan vuosikertomus 2011. Luettavissa: [http://kela.fi/in/internet/liite.nsf/NET/120412154043AK/\\$File/Kela_toimintakertomus2011_FI.pdf?OpenElement](http://kela.fi/in/internet/liite.nsf/NET/120412154043AK/$File/Kela_toimintakertomus2011_FI.pdf?OpenElement). Luettu: 24.6.2012.

Kela. 2012c. Kelan tilastot. Luettavissa: <http://kela.fi/in/internet/suomi.nsf/NET/011001094800TL?OpenDocument>. Luettu: 24.6.2012.

Kelanetti. 2012. Kelan intranet-sivusto. Luettu 9.7.2012.

Paananen, J. 2003. Tietotekniikan peruskirja. Docendo Finland Oy. Jyväskylä.

Pohjonen, R. 2002. Tietojärjestelmien kehittäminen. Docendo Finland Oy. Jyväskylä.

Rinta-Paavola, M. 2012. Toimeksiantajan edustaja, kehittämispäällikkö. Kansaneläkelaitos. Haastattelut ja palaverit opinnäytetyöprosessin aikana. Helsinki.

SAS Institute Oy. 2012. Yrityksen kotisivut. Luettavissa:
<http://www.sas.com/offices/europe/finland/>. Luettu: 27.1.2012.

Savolainen, S. 2011. James-menetelmä. Kelan sisäinen menetelmäinfotilaisuus 6.5.2011. Helsinki.

Stenberg, M. 2006. Tietojohtamisen arkkitehtuurit. Kustannusosakeyhtiö Otava. Helsinki.

Turtia, H. 2011. JAMES – iteratiivinen kehitys. Kelan sisäinen menetelmäinfotilaisuus 6.5.2011. Helsinki.

Vilkka, H., Airaksinen, T. 2003. Toiminnallinen opinnäytetyö. Kustannusosakeyhtiö Tammi. Helsinki.

Virkki, P., Somerniemi, A. 1999. Systeemityö tutuksi. 6. painos. Provano Oy. Vantaa.

Liitteet

Liite 1. Opinnäytetyössä käytetyt termit ja käsitteet

Aloitusvaihe (JAMES-menetelmässä)

JAMES-menetelmään liittyvä elinkaarivaihe, jonka aikana kiinnitetään projektin tavoitteet ja luodaan yhteinen käsitys siitä, mitä ollaan tekemässä. Aloitusvaiheessa suoritetaan myös tietojärjestelmään liittyvää esitutkimusta. (Savolainen 2001.)

Elinkaarivaihe (JAMES-menetelmässä)

JAMES-menetelmään liittyvä rakenneos, joka vaiheistaa JAMES-projektin. Elinkaarivaiheet ovat: Aloitusvaihe, Tarkennusvaihe, Rakennusvaihe ja Siirtymävaihe. Ne voivat sisältää yhden tai useampia iteraatioita, ja ne seuraavat toisiaan kronologisessa järjestyksessä. (Savolainen 2011.)

Inkrementaalisuus, inkrementti

Inkrementaalisuudella tarkoitetaan sitä, että tietojärjestelmäkehityksen lopputulosta valmistellaan pienissä osissa lisäyksittäin sen sijaan, että pyritäisiin tekemään valmis lopputulos yhdellä kerralla. Valmistunutta osakokonaisuutta kutsutaan inkrementiksi. Inkrementaalisuus liittyy kiinteästi RUP-pohjaisiin systeemityömalleihin. (Savolainen, 2011.)

Iteraatio

Iteratiivisessa systeemityömenetelmässä toteutettava kierros, joka sisältää yhden osan suunnittelun, riskianalyysin, toteutuksen sekä asiakkaan suorittaman arvioinnin. Iteraation lopputulos toimii seuraavan iteraation syötteenä, ja iteraatioita suoritetaan peräkkäin niin monta, että päästään haluttuun lopputulokseen (tai todetaan riskien kasvaneen niin suuriksi, että kehitystyö on järkevintä keskeyttää). (Savolainen, 2011.)

Iteratiivisuus

Erityisesti RUP-pohjaisessa systeemityössä toteutettava lähestymistapa, jossa suunnittelua, riskianalyysiä, toteutusta ja asiakkaan suorittamaa arviointia tehdään pienissä toistuvissa osissa jatkuvasti tarkentaen. (Savolainen, 2011.)

JAMES-käsikirja

Kelan intranettiin tallennettu ohjeisto JAMES-menetelmän käyttöön. Käsikirja sisältää myös JAMES-lopputuotteiden mallipohjat. (Turtia 2011.)

JAMES-menetelmä

Toimeksiantajaorganisaatiossa kehitetty RUP-pohjainen systeemityömalli, joka perustuu iteratiivisuuteen ja inkrementaalisuuteen. JAMES-menetelmän mukaisesti toteutettuja projekteja kutsutaan JAMES-projekteiksi. (Savolainen, 2011.) Opinnäytetyön työstämisen aikaan käytössä oli versio 3.1.

JAMES-tuki

Kelan IT-osastolla toimiva virtuaaliorganisaatio, joka vastaa JAMES-menetelmään liittyvästä kehitys-, tiedotus-, neuvonta- ja koulutustyöstä (Turtia 2011).

JAVA

C-kielestä kehitetty ohjelmointikieli, jolla voidaan tehdä erilaisia alustariippumattomia sovelluksia (Paananen 2003, 493).

Järjestelmän yleiskuvaus -dokumentti

JAMES-projekteissa tuotettava lopputuote, johon kuvataan kehitettävän järjestelmän määrittelyperusteet, yleiset toiminnallisuudet ja järjestelmäarkkitehtuuri. Dokumentti toimii järjestelmän käsikirjana projektin päätyttyä. (JAMES-käsikirja 2012.)

Kela, Kansaneläkelaitos

Julkisoikeudellinen laitos, joka vastaa Suomessa asuvien ihmisten perusturvasta. Kelasta haetaan ja sieltä maksetaan erilaisia etuuksia, esimerkiksi lapsilisää, opintotukea ja kansaneläkettä (Kela 2012b). Tämän opinnäytetyön toimeksiantajaorganisaatio.

Lopputuote (JAMES-menetelmässä)

Tietojärjestelmien kehitysprojekteihin liittyviä, JAMES-menetelmässä määriteltyjä tuotoksia, jotka projektin tehtävissä tulisi tuottaa. Lopputuotteet voivat olla esimerkiksi määrittelydokumentteja, kaavioita tai koodeja. (Turtia 2011.)

Lähtöaineisto (JAMES-menetelmässä)

JAMES-projekteissa tuotettuja lopputuotteita, jotka on huomioitava jossakin seuraavassa tehtävässä, kutsutaan lähtöaineistoksi. Lähtöaineisto voi olla pakollinen tai valinnainen. (Turtia 2011.)

Mainframe

Mainframella tarkoitetaan keskustietokoneympäristöä. Siinä varsinainen tietojen käsittely tapahtuu yhdellä tai useammalla suurella kapasiteetilla varustetulla keskustietokoneella. Niitä käytetään useilta tietokonepäätteiltä. (Paananen 2003, 25–26.)

Makro

Jonkin sovelluksen ohjelmoinnissa laadittu komentojen joukko. Makro tallennetaan usein erikseen, ja makroa voidaan kutsua makrokutsulla suorittamaan sille määriteltäviä tehtäviä nopeuttaen rutiinitehtäviä. (Paananen 2003, 495.)

Metadata, metatieto

Metadatala tarkoitetaan tietoa tiedosta, eli se on eräänlaista kuvailutietoa. Tyypillistä metadataa on esimerkiksi tiedon nimi, tietotyyppi ja tiedon pituus. Organisaation metadataa voidaan kerätä metatietokantaan tai tietohakemistoon. (Hovi, Ylinen & Koistinen 2001, 110–111).

Prototyypilähestymistapa, prototyypimalli

Vaihejakomalli, jossa asiakkaan alustavien toiveiden perusteella rakennetaan prototyyppi, jonka avulla järjestelmän ominaisuuksia demonstroidaan asiakkaalle. Demonstraatiosta saadun palautteen perusteella prototyyppiä parannellaan. Tätä prototyypin parantelu – demonstraatio –kiertoa jatketaan, kunnes asiakas on tyytyväinen prototyyppiin. Lopuksi viimeisimmän prototyypin perusteella toteutetaan lopullinen järjestelmä. (Pohjonen 2002, 41.)

RUP

Rational Unified Process on hyväksi havaituista käytännöistä koostettu ohjelmistokehityksen prosessikehys. Sen perusajatuksia ovat iteratiivisuus ja inkrementaalisuus. (IBM Rational Unified Process, 2011.)

Sanasto-dokumentti

JAMES-projekteissa tuotettava lopputuote, johon kootaan projektissa käytettävät keskeiset termit ja lyhenteet selkokielisine selitteineen (JAMES-käsikirja 2012).

SAS, SAS-teknologia

SAS Institute Inc. on yhdysvaltalainen liiketoiminta-analytiikkaan keskittynyt yritys. Sen päätuotteita ovat SAS-ohjelmistot, joiden päätarkoituksena on taulukkomuotoisen datan käsittely ja hyödyntäminen esimerkiksi liiketoiminta- ja tutkimustarkoituksiin. SAS-ohjelmointia varten on kehitetty SAS-komentokieli, jolla voidaan esimerkiksi käsitellä tietomassoja ohjelmallisesti ja tuottaa erilaisia raportteja ja tulosteita. SAS on lyhenne sanoista Statistical Analysis System. (SAS Institute Oy 2012.) Toimeksiantajaorganisaatioissa SAS-teknologialla tarkoitetaan tilastointi- ja raportointijärjestelmiin liittyvän kehitystyön toteuttamista SAS Institute Inc.:n tuotteita tai SAS-komentokieltä hyödyntäen (Rinta-Paavola 2012).

Sidosryhmien tarpeet -dokumentti

JAMES-projekteissa tuotettava lopputuote, johon kootaan projektin sidosryhmien tarpeet kehitettävälle järjestelmälle (JAMES-käsikirja 2012).

Spiraalimalli

Vaihejakomalli, jossa koostuu suunnittelu-, riskianalyysi-, tuotanto- ja arviointivaiheista. Näitä vaiheita toistetaan spiraalimaisesti niin kauan, että järjestelmä saadaan valmiiksi (tai riskianalyysivaiheen perusteella tullaan siihen tulokseen, ettei kehitystyötä kannata jatkaa). (Pohjonen 2002, 42.)

Systeemitestisuunnitelma-dokumentti

JAMES-projekteissa tuotettava lopputuote, jossa esitellään kehitettävän järjestelmään liittyvät testausmenettelyt (JAMES-käsikirja 2012).

Systeemyö (ks. tietojärjestelmien kehittäminen)

Systeemyö on tietojärjestelmän tai ohjelman rakentamista, kunnossapitamista ja kehittämistä (Virkki & Somerniemi 1999, 171). Sitä sovelletaan käytäntöön erilaisten menetelmien ja mallien avulla (Pohjonen, 2002, 68).

Tehtävä (JAMES-menetelmässä)

JAMES-menetelmään liittyvä osakokonaisuus, jonka aikana tietyt roolit suorittavat tiettyjä asioita ja tuottavat tehtävälle määriteltyjä lopputuotteita (Turtia 2011).

Tietoarkkitehtuuri

Tietoarkkitehtuuri on osa tietohallinnon johtamista ja kehittämistä. Siinä otetaan kantaa tiedon hallintaan keskityksen ja hajautuksen sekä tehtäväkohtaisten ja yhteiskäyttöisten tietojen suhteen. Tarkoituksena on kuvata nykytila ja tavoitetila sekä tehdä tarvittavia linjauksia ja ohjeistuksia. (Stenberg, 2006, 72.)

Tietohakemisto

Tietokanta tai dokumentti, johon on tallennettu kuvaukset kaikista tiettyyn tietojärjestelmään sisältyvistä tiedoista eli metadataa. Kuvattuina ovat muun muassa tietojärjestelmän syötteet, tulosteet, tietovarastojen rakenteet sekä tietoelementtien tarkat määrittelyt. (Pohjonen 2002, 123.) Toimeksiantajaorganisaatiossa tietohakemiston tietokantaan tallennetaan organisaatiossa käsiteltävien tietojen metadataa ja kuvaustietoja (Rinta-Paavola 2012).

Tietojärjestelmien kehittäminen (ks. systeemyö)

Määrämuotoista, usein projektiluonteista toimintaa, jonka tavoitteena on toimintaa tukevien uusien tietojärjestelmien kehittäminen tai uusien palveluiden ja tuotteiden luominen (Stenberg 2006, 103). Kehitysryhmän suorittama prosessi, joka tähtää tiettyssä

ympäristössä sijaitsevan kohdejärjestelmän muutokseen asetettujen tavoitteiden mukaisesti (Pohjonen 2002, 15).

Tietojärjestelmä

Tietojen käsittelyyn liittyvä kokonaisuus, joka koostuu laitteista, ohjelmista, tallennetuista tiedoista, järjestelmän käyttäjistä sekä käyttöön liittyvistä ohjeistuksista (Hovi ym. 2001, 18).

Tietoturvaselvitys-dokumentti

JAMES-projekteissa tuotettava lopputuote, johon kirjataan kehitettävälle järjestelmälle asetettavat tietoturva-vaatimukset sekä niihin liittyvät toimenpiteet (JAMES-käsikirja 2012).

Toiminnallinen opinnäytetyö

Usein työelämälähtöinen opinnäytetyö, jonka tavoitteena on jonkin käytännön toiminnan ohjeistaminen, opastaminen, toiminnan järjestäminen tai järjeistäminen (Vilka & Airaksinen 2003, 9).

Työvaihe (JAMES-menetelmässä)

JAMES-menetelmään liittyvä rakenneosa. Työvaiheita ovat: esitutkimus, määrittely, suunnittelu, toteutus, testaus, käyttöönotto, kokoonpano- ja muutoshallinta, projektin hallinta sekä ympäristöt. Näitä vaiheita toteutetaan elinkaarivaiheiden aikana koko projektin ajan erilaisilla painotuksilla. (Savolainen 2011.)

Vaihejako

Yleinen tapa jakaa tietojärjestelmien kehittäminen ja systeemyö peräkkäisiin vaiheisiin on seuraava: esitutkimus-, vaatimusmäärittely-, järjestelmäanalyysi-, suunnittelu-, toteutus-, testaus-, käyttöönotto- ja ylläpitovaihe (Pohjonen 2002; 26). Vaihejaosta käytetään myös termiä tietojärjestelmän elinkaari, jolloin tarkoitetaan aikaa, joka alkaa tietojärjestelmän kehittämisestä ja päättyy sen poistamiseen käytöstä (Haikala & Märijärvi 2004, 36).

Vaihejakomalli

Tietojärjestelmien kehittämisprosessin pohjana toimiva malli, joka sisältää kaikki olennaiset systeemityövaiheet ja kuvaa niiden suoritusjärjestyksen. Vaihejakomallista käytetään usein myös termiä elinkaarimalli. (Pohjonen 2002, 39.)

Vesiputousmalli

Vaihejakomalli, jossa systeemityön vaiheet seuraavat toisiaan kronologisessa järjestyksessä. Prosessi etenee pääsääntöisesti eteenpäin, ja palaamista edellisiin vaiheisiin pyritään välttämään. (Pohjonen 2002, 40.)

Visio-dokumentti

JAMES-projekteissa tuotettava lopputuote, johon kirjataan asiat, jotka projektissa pyritään saavuttamaan. Visio auttaa varmistamaan, että kaikilla projektin toimijoilla on yhteinen käsitys siitä, mitä ollaan kehittämässä ja miksi. (JAMES-käsikirja 2012.)

28.9.2012 22:26:00

Liite 2. JAMES-lopputuote Sidosryhmän tarpeet

Sidosryhmien tarpeet

Tietohakemiston tekstejä hyödyntävä tietojärjestelmä

Vastuu: IT-osasto / Järjestelmäkehitysryhmä / Miia Nopanen

Versionhallinta

Versio	Päiväys	Muokkaaaja	Kommentit
0.1	24.02.2012	Miia Nopanen	Aloitettu dokumentin kokoaminen haastatteluiden pohjalta => katselmoitavaksi projektin omistajalle
0.2	12.03.2012	Miia Nopanen	Tehty omistajan esittämät korjauksia => katselmoitavaksi sidosryhmien edustajille
0.3	22.03.2012	Miia Nopanen	Sisältökatselmoinnista saadut korjausedotukset tehty
0.4	03.04.2012	Miia Nopanen	Rakennekatselmoinnista saadut korjausedotukset tehty
0.5	16.04.2012	Miia Nopanen	Sisältökatselmoinnista saadut korjausedotukset tehty
1.0	08.07.2012	Miia Nopanen	Toimeksiantajan hyväksymä versio seuraavien iteraatioiden käyttöön

Dokumentin tiedot

Valmistumispäivä
Sijainti

08.07.2012

Tulostettu
Mallipohjan versio

C:\Users\Miia\Desktop\Opinnäytetyö\Aloitusvaihe\Oparin liitteiksi\1_Sidosryhmien tarpeet v1.0.docx

28.9.2012 22:26

JAMES 3.1

Sisältö

1 Johdanto	5
1.1 Tarkoitus	5
1.2 Laajuus	5
1.3 Määritelmät ja lyhenteet.....	6
1.4 Viittaukset.....	6
1.5 Yleiskuvaus.....	6
2 Sidosryhmä 1; SAS-teknologialla toteutettuja tilastointi- ja raportointijärjestelmiä kehittävät ja ylläpitävät henkilöt	7
2.1 Yleiskuvaus.....	7
2.2 Yhteyshenkilöt	7
2.3 Tarpeet.....	7
2.3.1 Ensisijaiset tarpeet	8
2.3.2 Toissijaiset tarpeet	10
2.3.3 Selainpohjaiseen raportointijärjestelmään liittyvät tarpeet	10
3 Sidosryhmä 2; Tilastollisen materiaalin loppukäyttäjät	12
3.1 Yleiskuvaus	12
3.2 Yhteyshenkilöt	12
3.3 Tarpeet	12
3.3.1 Ensisijaiset tarpeet	13
3.3.2 Toissijaiset tarpeet	13
4 Sidosryhmä 3; SAS-sovellusarkkitehtuurin ja tietoarkkitehtuurin kehittäjät	14
4.1 Yleiskuvaus	14
4.2 Yhteyshenkilöt	14
4.3 Tarpeet	14
4.3.1 Ensisijaiset tarpeet	15

4.3.2 Toissijaiset tarpeet	16
5 Sidosryhmä 4; Tiedotettavat tahot	17
5.1 Yleiskuvaus	17
5.2 Yhteyshenkilöt	17
5.3 Tarpeet	17
5.3.1 Ensisijaiset tarpeet	17
5.3.2 Toissijaiset tarpeet	18
LIITE 1. Sidosryhmähaastatteluiden kysymykset	19

1 Johdanto

1.1 Tarkoitus

Kerää sidosryhmien tarpeet -tehtävä on JAMES-menetelmän mukaisten projektien esituskimus-työvaiheeseen liittyvä tehtävä. Sen tarkoituksena on kartoittaa tulevaa tietojärjestelmää käyttävät sidosryhmät sekä niiden projektille osoittamat odotukset ja tarpeet. Tehtävän lopputuotteena tuotetaan Sidosryhmien tarpeet -dokumentti, johon sidosryhmiltä kerätyt tarpeet dokumentoidaan ja niistä muodostetaan myöhemmin järjestelmälle kohdistetut vaatimukset.

Dokumentissa kerrotaan myös, miten sidosryhmien tarpeet on kerätty ja miten tarpeita on prosessin edetessä jatkokehitetty. Kaikki sidosryhmien tuottamat tarpeet kirjataan ylös tähän dokumenttiin. Tarpeista tullaan muokkaamaan tulevan järjestelmän vaatimukset. Vaatimukset priorisoidaan, ja projektissa toteutetaan toiminnallisuudeltaan melko yksinkertainen ja kapea-alainen ratkaisu huomioiden vain korkeimman prioriteetin vaatimukset. Tulevissa projekteissa ratkaisua voidaan sitten laajentaa kattamaan prioriteetiltään pienempiäkin vaatimuksia.

1.2 Laajuus

Tämä dokumentti liittyy Tietohakemiston tekstejä hyödyntävien tietojärjestelmien kehitysprojektiin. Dokumentti keskittyy projektin osatehtävään, jonka tavoitteena on kehittää SAS-teknologialla toteutettava, Tietohakemiston tietokannasta, eräajoista, makrosta, makrokutsuista, ohjeistuksista sekä tilastointi- ja raportointijärjestelmien kanssa työskentelevistä toimihenkilöistä koostuva tietojärjestelmä. Tietojärjestelmän alustavana toiminta-ajatuksena on, että SAS-teknologialla toteutettujen tilastointi- ja raportointijärjestelmien kehittäjät ja ylläpitäjät voisivat projektissa kehitettävän makron avulla hyödyntää Tietohakemistoon tallennettuja kuvaustekstejä omissa SAS-ohjelmissaan. SAS-teknologia sisältää ominaisuuden, jolla voidaan tallentaa sekä data että metadata (selitteet ja tiedonkuvaukset) tiedostoille ja sen muuttujille. Tietohakemiston tekstejä hyödyntävä tietojärjestelmä-projekti kuuluu laajemmin meneillään olevaan tietoarkkitehtuurin kehittämiseen, jonka tavoitteena on kehittää tiedonhuoltoprosessia ja parantaa organisaatiossa käytettävien tietojen laatua.

Tilastointi- ja raportointijärjestelmiin liittyvissä projekteissa ei ole aikaisemmin koestettu tai hyödynnetty JAMES-menetelmää. Tämä projekti dokumentaatioineen toimii kokeiluna ja pilottina, kuinka JAMES-menetelmää voitaisiin jatkossa hyödyntää vastaavanlaisissa kehitysprojekteissa. Jotta alun perin Java-projekteille kehitetty JAMES-menetelmä saataisiin soveltumaan mahdollisimman hyvin myös tilastointi- ja raportointijärjestelmiin liittyviin projekteihin, on dokumentin tuottamisessa konsultoitu myös JAMES-tukea.

Tässä projektissa tarpeet ja vaatimukset kirjataan tähän Sidosryhmien tarpeet -dokumenttiin. Projektissa ei hyödynnetä toimeksiantajan käytössä olevia valmisohjelmistoja tarpeiden kirjaamiseen, vaatimusten mallintamiseen ja generointiin.

Sidosryhmien tarpeet -dokumentti toimii projektin lähtömateriaalina ja lähtökohtana seuraaville dokumenteille:

- Visio

1.3 Määritelmät ja lyhenteet

Tämän dokumentin ymmärtämisen kannalta olennaiset termit ja lyhenteet on kirjattu Sanasto-dokumenttiin.

1.4 Viittaukset

Tässä vaiheessa projektia ei ole käytettävissä taustamateriaalia.

1.5 Yleiskuvaus

JAMES-tuen kanssa käydyn sähköpostikeskustelun perusteella alkuperäisen dokumenttipohjan rakennetta on hieman muokattu, sillä alkuperäinen rakenne ei oikein soveltunut useiden sidosryhmien ja niiden tarpeiden kuvaamiseen. Dokumentin kappaleessa 1 kuvataan dokumentin taustatekijöitä ja liitoksia. Kappaleesta 2 alkaen esitellään kukin sidosryhmä omassa pääkappaleessaan, ja niiden alle on eroteltu kyseisen sidosryhmän tarpeet ensisijaisiin sekä toissijaisiin tarpeisiin.

Sidosryhmien tarpeiden kartoitusprosessi aloitettiin keskustelemalla projektin omistajan kanssa. Hän kertoi näkemyksensä projektin käynnistämisen syistä ja tarpeellisuudesta, sekä määritteli projektin kohteena olevan tietojärjestelmän sidosryhmät ja näiden edustajat. Tarpeiden kartoitus päätettiin suorittaa yksilöhaastatteluin, jotta jokaisen sidosryhmän edustaja saisi varmasti näkemyksensä esille. Ensin laadittiin alustavat kysymykset, joita projektin omistajan katselmoinnin jälkeen vielä hieman tarkennettiin ja muokattiin (ks. liite 1.). Kysymykset laadittiin sellaisiksi, että ne tukevat myös tämän Sidosryhmien tarpeet -dokumentin laatimista.

Haastatteluita suoritettiin 8 kappaletta 4 viikon aikana. Haastatteluiden jälkeen sidosryhmien tarpeet kirjattiin tähän dokumenttiin siten, että ne jaettiin ensi- ja toissijaisiin tarpeisiin dokumentin laatijan toimesta. Tämän jälkeen kaikkien sidosryhmien edustajat katselmoivat sähköpostitse haastatteluiden perusteella kirjatut tarpeet ja hyväksyivät jaon ensi- ja toissijaisiin tarpeisiin. Katselmoinnista saatujen kommenttien perusteella joitakin sanamuotoja vielä tarkennettiin.

2 Sidosryhmä 1; SAS-teknologialla toteutettuja tilastointi- ja raportointijärjestelmiä kehittävät ja ylläpitävät henkilöt

Tästä sidosryhmästä haastateltiin 4:ää eri tiimeissä tai tehtävissä työskentelevää henkilöä. He edustavat yhteensä noin 25:tä sidosryhmäänsä kuuluvaa henkilöä. Tästä ryhmästä haastateltiin useita henkilöitä sen vuoksi, että he edustavat uuden tietojärjestelmän käyttäjiä.

2.1 Yleiskuvaus

Tämän sidosryhmän edustajat työskentelevät IT-osastolla. Heidän ja edustamansa ryhmän työtehtäviin kuuluu mm. SAS-teknologialla toteutettujen tilastointi- ja raportointijärjestelmien kehitys ja ylläpito. Sidosryhmään 1 kuuluvat henkilöt ovat sellaisia, jotka tulevat nimenomaan käyttämään tulevaa tietojärjestelmää ja siihen kuuluvaa SELITE-makroa omissa ohjelmissaan.

Osa henkilöistä on tottuneita käyttämään makroja ohjelmointityössään, osa taas ei.

2.2 Yhteyshenkilöt

- Esimies; mm. eläkkeitä sekä vammais- ja kuntoutusetuuksia koskevia tilastointi- ja raportointijärjestelmiä ylläpitävien henkilöiden esimies IT-osaston asiakasvastuuryhmässä
- IT-asiantuntija; tietovaraston kehittäminen, tilastointi- ja raportointijärjestelmien kehittäminen ja ylläpito
- IT-asiantuntija; takaisinperinnän tilastointijärjestelmät, lääketilastointi
- Esimies; mm. perhe-etuuksia, työttömyysturvaa ja sairasturvaa koskevia tilastointi- ja raportointijärjestelmiä ylläpitävien henkilöiden esimies IT-osaston asiakasvastuuryhmässä, selainpohjaisen raportointijärjestelmän liittymien asiantuntija (mahdollinen tuleva käyttäjäryhmä)

2.3 Tarpeet

Sidosryhmän 1 edustajat näkivät Tietohakemiston tekstejä hyödyntävän tietojärjestelmän kehittämisen tarpeelliseksi ja merkittäväksi uudistukseksi tilastointi- ja raportointijärjestelmien alueella. Näihin järjestelmiin liittyvien tiedostojen ja muuttujien selitteet on nykyään hyvin pitkälti tuotettu kirjoittamalla selitteet ohjelmakoodiin mukaan. Tällä tavalla hallittavien selitteiden ylläpito ja päivitys on työlästä, sillä koodattu ohjelma joudutaan siirtämään ohjelmakehityksessä muokattavaksi muutosympäristöön, jotta päivitys voidaan tehdä. Lisäksi päivitys on tehtävä yksi kerrallaan jokaiseen ohjelmaan, mikä on hyvin työlästä ja aikaa vievää. Joskus selitteiden kirjaaminen jääkin kokonaan tekemättä kiireen vuoksi.

Uusi tietojärjestelmä mahdollistaisi selitteiden tuottamisen ohjelmallisesti Tietohakemistosta, mikä helpottaisi ja nopeuttaisi tilastointi- ja raportointijärjestelmien ylläpitoa suuresti. Lisäksi se edistäisi standardien mukaista menettelytapaa selitteiden tuottamisessa. Lisäksi tilastointitiedostojen siirtäminen pääkoneympäristöstä palvelinympäristöön asettaa omat vaatimuksensa tiedostojen sisällölle. Tiedostojen ja muuttujien selitteiden on tällöin oltava kunnossa, jotta niiden omatoimikäyttö olisi selkeää loppukäyttäjille.

Tässä yhteydessä haastateltiin myös selainpohjaisen raportointijärjestelmän näkökulman edustajaa, vaikka edellä mainittuun järjestelmään liittyvää toiminnallisuutta ei tulla toteuttamaan projektin tässä vaiheessa. Joitakin tarpeita selvitettiin kuitenkin jo nyt, ja ne on kirjattu alakappaleeseen 2.3.3.

Seuraavissa alakappaleissa esitellään yksityiskohtaisemmin sidosryhmän 1 odotukset ja tarpeet jaoteltuna ensisijaisiin ja toissijaisiin tarpeisiin.

2.3.1 Ensisijaiset tarpeet

2.3.1.1 Tietojärjestelmän käytön on oltava yksinkertaista

Tietojärjestelmän ja siihen kuuluvan makro-ohjelman tulevien käyttäjien makrojen käyttö-osaaminen vaihtelee jonkin verran. Tästä johtuen makron käyttämisen tulee olla yksinkertaista, jotta se ei jää käyttämättä hankaluutensa vuoksi.

2.3.1.2 Tietojärjestelmän on toimittava määritysten mukaisesti oikein

Tietojärjestelmän tulee tuottaa käyttäjille oikeita tietoja, jotta sitä voidaan käyttää. Virheelliset tiedot vievät järjestelmän uskottavuuden ja voivat aiheuttaa ongelmia muissa tätä järjestelmää hyödyntävissä tietojärjestelmissä.

2.3.1.3 Makro-ohjelmakoodia koskevia standardeja on noudatettava

Makro tulee tallentaa sovittuun paikkaan tuotantoympäristössä, jotta sen käyttöönotto sujuu standardeja noudattaen. Makron ohjelmakoodi versioidaan ja sen sijoituspaikka on mainittava myös ohjeistuksissa.

2.3.1.4 Tietojärjestelmän käyttöönotosta ja toiminnasta on laadittava selkeä ohjeistus

Tietojärjestelmästä ja siihen sisältyvän makron käyttämisestä on laadittava ohjeistus, josta tulee ilmi ainakin seuraavat asiat:

- Makron kutsuminen toisesta ohjelmasta; mitä muutoksia vaati jo olemassa oleviin ohjelmiin
- Makron tuottamat tiedot ja niiden hyödyntäminen kutsuvassa ohjelmassa
- Mahdollisesti myös kuvaus koko tietojärjestelmän toiminnasta sekä makron tarkemmasta toiminnallisuudesta

2.3.1.5 Tietojärjestelmän käyttöönotosta ja toiminnallisuudesta on informoitava käyttäjiä esimerkiksi tiedotustilaisuudessa

Jotta mahdollisimman moni tilastointi- ja raportointijärjestelmien kehittäjistä ja ylläpitäjistä saisi tiedon ja käsityksen uudesta tietojärjestelmästä, on syytä järjestää pienimuotoinen

tiedotustilaisuus. Tilaisuudessa olisi hyvä esitellä demon avulla erityisesti makron toimintaa, sillä se on tietojärjestelmän osa, jota käyttäjät erityisesti tulevat hyödyntämään työssään.

Voisi olla tarpeen järjestää myös hieman laajempi tiedotustilaisuus, jossa esiteltäisiin muitakin käytössä olevia makroja ja perehdytettäisiin makroajatteluun perusteellisemmin.

2.3.1.6 Tietojärjestelmä ja makro on suojattava muokkaamiselta

Tietojärjestelmä ja makron ohjelmakoodi on syytä suojata siten, että niitä pääsee vain käyttämään, ei muokkaamaan. Tällä estetään se, ettei joku vahingossa käy muuttelemassa tietojärjestelmää tai makroa ja riko niiden toiminnallisuutta. Samalla on mietittävä, kenelle annetaan päivitysoikeudet eli ylläpitovastuu.

2.3.1.7 Tietojärjestelmän purkuajovaihe on ajastettava säännöllisesti, mutta purun on onnistuttava tarpeen vaatiessa myös muulloin

Tietojärjestelmään ajastetaan lähdejärjestelmän tietokantojen purku säännöllisesti (esim. 1 kerta / kk lähellä kuun vaihdetta, tai 1 krt / vko maanantai-aamuyöllä). Purkuajojen ajastamisessa on huomioitava pääkoneen kuormitustilanne, ja ajoittaa se kuormitushuippujen ulkopuolelle. Tarpeen vaatiessa purkuajot on voitava ajaa erillisellä tilauksella.

2.3.1.8 Järjestelmän tulee voida tuottaa Tietohakemistosta myös vanhempien kuvausversioiden tietoja aikaparametrilla

Yleensä tarvitaan Tietohakemistoon tallennetun kuvaustekstien uusinta versiota. Joskus voi kuitenkin tulla tilanne, että tarvitaankin jonkin menneen ajankohdan eli vanhemman version tietoja, jolloin tekstit tulee voida hakea esimerkiksi aikaparametrin avulla.

2.3.1.9 Makroa kutsuva ohjelma ei saa pysähtyä virheeseen, vaikka makro ei onnistuisikaan tuottamaan kaikkia käyttäjän kutsuun määrittelemiä selitteitä

Jos makro ei syystä tai toisesta onnistukaan tuottamaan käyttäjän määrittelemiä selitteitä, se ei saa aiheuttaa kutsuvan ohjelman kaatumista. Puutteellisista tiedoista on kuitenkin huolehdittava informaatio käyttäjälle, jotta tämä voi reagoida tilanteeseen. Tiedotettavia tahoja ovat myös makron / tietojärjestelmän ylläpitäjät. Mietittävä, voisiko esim. käytössä olevaa IT-palveluiden hallintajärjestelmän toiminnallisuutta hyödyntää tässä.

2.3.1.10 Makro on toteutettava siten, ettei käyttäjän tarvitse kirjoittaa makrokutsuun parametreiksi kaikkia teknisiä nimiä erikseen

Joissakin tilastointi- ja raportointijärjestelmissä saatetaan käsitellä kymmeniä eri muuttujia. Olisi hyvä, jos käyttäjän ei tarvitsisi itse kirjata kaikkien näiden muuttujien teknisiä nimiä makrokutsuun. Voisiko tässä hyödyntää esimerkiksi SAS-ohjelmointikielen proc contents -proseduurin tuloksia?

2.3.2 Toissijaiset tarpeet

2.3.2.1 Voiko uutta tietojärjestelmää hyödyntää jotenkin myös uuden ETL-välineen kanssa?

Tilastointi- ja raportointijärjestelmien ylläpitoon on tulossa uusi ETL-väline. Olisiko mahdollista, että kehitettävä tietojärjestelmä voisi jotenkin tuottaa tietoja EDW:hen, jonne SAS:illa ei ole pääsyä tällä hetkellä?

2.3.2.2 Ohjelmassa tuotetuille uusille muuttujille on saatava myös selitteet

Tilastointijärjestelmissä luodaan purkutiedostosta tuotettujen muuttujien lisäksi niin kutsuttuja jalostettuja muuttujia. Tällä hetkellä näiden uusien muuttujien selitteet on mahdollista tuottaa ohjelmallisesti pääkoneympäristön tekstitiedostomenettelyn kautta. Kuinka jatkossa menetellään näiden uusien luotujen muuttujien selitteiden kanssa, saadaanko nekin tuotettua Tietohakemistosta?

2.3.2.3 Tietojärjestelmää tulee voida käyttää pääkoneympäristössä, SAS/EG:llä ja MikroSAS:lla

Järjestelmän voisi aluksi ottaa käyttöön pääkonepuolella, mutta sille olisi tarvetta myös työaseman puolella eli sovelluskehityksessä, jossa käytetään SAS/EG- ja MikroSAS-välineitä.

2.3.3 Selainpohjaiseen raportointijärjestelmään liittyvät tarpeet

2.3.3.1 Selainpohjaiseen raportointijärjestelmään liittyvien toiminnallisuuksien koodaaminen on suoritettava WebFOCUS-ohjelmalla

Raportit ovat dynaamisia raportteja, joiden avulla loppukäyttäjät voivat tutkia Kelan etuus-tietoja summatasolla. Raportointijärjestelmä on kehitetty WebFOCUS -nimisellä sovellus-tuotteella, ja myös Tietohakemistosta tuotettavia tekstejä suorittava yleisohjelma on koo-dattava WebFOCUS:lla.

2.3.3.2 Selainpohjaiseen raportointijärjestelmään liittyvän toiminnallisuuden on oltava "normaalipalvelu"

Uuden yleisosan käyttäminen ei saa sisältää ylimääräisiä välivaiheita, vaan sovelluskehittäjien työn on oltava yhtä vaivatonta kuin aiemminkin.

2.3.3.3 Selainpohjaiseen raportointijärjestelmään liittyvässä toiminnallisuudessa on otettava huomioon tila-asiat

Järjestelmään liittyen on huomioitava fyysiset tilarajoitteet, kuinka paljon tekstiä voidaan kirjoittaa ja siirtää kerrallaan.

2.3.3.4 Etuustilastointiin ja raportointiin liittyvät yhteiset osat ovat tietyn henkilön vastuulla

Vastuuhenkilöä on tiedotettava kehitystyöstä ja tulevasta järjestelmästä.

3 Sidosryhmä 2; Tilastollisen materiaalin loppukäyttäjät

Tästä sidosryhmästä haastateltiin 2:ta eri tiimeissä tai tehtävissä työskentelevää henkilöä. He edustavat yhteensä noin 20:tä sidosryhmäänsä kuuluvaa henkilöä.

3.1 Yleiskuvaus

Tämän sidosryhmän henkilöt työskentelevät pääsääntöisesti muilla osastoilla kuin IT-osastolla. Sidosryhmän 2 henkilöt toimivat usein toimeksiantajan roolissa, eli he tilaavat IT-osastolta tutkimuskäyttöön tarvittavia tilastointitiedostoja. He eivät itse todennäköisesti tule käyttämään tietojärjestelmää tai makroa, mutta tiedostojen loppukäyttäjinä he tulevat hyötymään tietojärjestelmän toiminnallisuuden tuloksista eli oikeista ja ajan tasaisista selitteistä.

Tästä johtuen tämän sidosryhmän edustajille esitettyjä kysymyksiä muokattiin siten, että niistä poistettiin selkeästi tekniikkaan ja järjestelmän toteutukseen liittyvät kysymykset. Kysymysten painopiste oli siinä, kuinka he hyödyntävät selitteitä työssään, mitä hyötyä niistä on, ja mitä ongelmia niiden puuttuminen saattaa aiheuttaa.

3.2 Yhteyshenkilöt

- Tutkija; tehtävänä erityisesti lääkekorvausasioihin liittyvien tiedostojen tarkistus ja tarvittaessa korjaus. Suorittaa tutkijoiden pyynnöstä myös analysointia.
- Aktuaariasiantuntija; tehtävänä erilaisten ennusteiden ja raporttien laatiminen sekä lainmuutoslaskelmien suorittaminen. Tiimissä kullakin henkilöllä on vastuullaan nimetyt etuudet, mutta joskus joudutaan kiiretapauksissa työskentelemään myös itselle entuudestaan tuntemattomien etuuksien ja aineistojen kanssa.

3.3 Tarpeet

Sidosryhmän edustajan mukaan erityisesti muuttujien selitteet ovat tarpeellisia, sillä niistä tutkitaan usein muuttujien sisältöä. Yleisesti käytössä olevat muuttujat ovat jo tulleet tutuiksi, mutta erityisesti lyhenteitä sisältävät selitteet tai erikoismuuttujien selitteet voivat olla vaikeita selvittää; asiasisältö pitää joko tietää tai etsiä tietuekuvauksista. Tietuekuvauksien löytäminen on joskus haasteellista, jos joutuu nopeasti perehtymään johonkin itselle entuudestaan tuntemattomiin etuuksiin ja aineistoihin. Dokumenttien etsimiseen verkko-levyjen kansioista kuluu aikaa.

Seuraavissa alakappaleissa esitellään yksityiskohtaisemmin sidosryhmän 2 odotukset ja tarpeet jaoteltuna tärkeisiin ja toissijaisiin tarpeisiin.

3.3.1 Ensisijaiset tarpeet

3.3.1.1 SAS-tiedostoilla ja -muuttujilla tulee olla selitteet

Selitteistä tutkitaan tiedostojen ja muuttujien sisältöä ja tarkoitusta. Jos selitteet puuttuvat, joudutaan tietoa etsimään tietuekuvauksista verkkolevyjen kansioista tai tiedustelemaan erikseen IT-osastolta, mikä voi olla aikaa vievää.

3.3.1.2 Selitteiden tulee olla ajan tasalla

Ajoittain muuttujiin ja niiden kuvauksiin tulee muutoksia. Jos näitä muutoksia ei ole päivitetty muuttujien selitteisiin, voi siitä seurata väärinkäsityksiä ja turhaa työtä.

3.3.1.3 Tietojärjestelmän käyttöönotosta ja yleisistä toiminnallisuuksista tulee tiedottaa myös kyseistä sidosryhmää

Vaikka sidosryhmään kuuluvat henkilöt eivät tulekaan itse käyttämään uutta tietojärjestelmää, heidän on hyvä saada tieto sen käyttöönotosta. Ajan myötä heille saattaa syntyä ajatuksia siitä, kuinka he voisivat paremmin hyödyntää uuden järjestelmän tuottamia toiminnallisuuksia.

3.3.2 Toissijaiset tarpeet

3.3.2.1 Selitteeseen tule mahdollisuuksien mukaan voida kirjata muutakin tietoa, kuin vain otsikkotieto

Usein pelkkä otsikko ei vielä kerro paljoakaan muuttujan sisällöstä. Olisi hyvä, jos selitteeseen saataisiin kirjattua otsikon lisäksi muutakin tietoa; esimerkiksi sukupuolta kuvaavan muuttujan selitteessä voisi olla mukana muuttujan mahdolliset arvot ja niiden selitykset. Joissakin tapauksissa selitteessä voisi olla vaikkapa viittaus johonkin paikkaan, josta voi käydä tutkimassa muuttujan tietoja tarkemmin ja etsiä lisätietoja.

3.3.2.2 Tietojärjestelmää tulee voida hyödyntää myös omatoimikäytössä tai Kelan ulkopuolelta tulevista tiedostoista

Joissakin tapauksessa tutkijat saattavat itse suorittaa aineistojen poimintoja ja luoda niistä tiedostoja. Heidän tulisi voida hyödyntää tietojärjestelmää omissa ohjelmissaan. Lisäksi joskus käytössä on Kelan ulkopuolelta tulevia tiedostoja (esim. Tilastokeskukselta). Olisi hyvä, jos näissäkin tiedostoissa voitaisiin hyödyntää tulevaa tietojärjestelmää.

4 Sidosryhmä 3; SAS-sovellusarkkitehtuurin ja tietoarkkitehtuurin kehittäjät

Tästä sidosryhmästä haastateltiin yhtä henkilöä. Hän edustaa yhteensä noin 5:tä sidosryhmäänsä kuuluvaa henkilöä.

4.1 Yleiskuvaus

Tähän sidosryhmään kuuluu henkilöitä ja tahoja, jotka pyrkivät tuomaan esille ja kehittämään SAS-sovellusarkkitehtuuriin sekä tietoarkkitehtuuriin liittyviä asioita omissa organisaatioryhmissään. He eivät itse tule varsinaisesti käyttämään tietojärjestelmää tai makroa, mutta heillä on omia arkkitehtuurien kehittämiseen liittyviä odotuksia ja tarpeita järjestelmälle.

4.2 Yhteyshenkilöt

- Esimies; SAS-sovellusarkkitehtuurin ja tietoarkkitehtuurinäkökulman edustaja. sovellusasiantuntijoiden esimies, projektin omistaja.

4.3 Tarpeet

SAS-sovellusarkkitehtuurin sekä tietoarkkitehtuurin kehittämistyöhön liittyy erilaisten linjauksen tekeminen. Esimerkiksi tällä hetkellä ei ole olemassa mitään linjausta, joka ottaisi kantaa selitetietojen käyttämiseen SAS-teknologialla tuotetuissa tiedostoissa. Se tiedetään, että siirryttäessä pääkoneympäristöstä palvelinympäristöön tiedostojen selitteiden on oltava kunnossa. Tällä hetkellä karkean arvion mukaan noin 30 %:ssa pysyvässäilytyksessä olevissa tiedostoissa selitteet ovat puutteellisia, ja muissa tiedostoissa jopa 60 % on vailla selitteitä.

Tarvittaisiin helppokäyttöinen menetelmä selitteiden lisäämiseksi SAS-tiedostoihin, jolloin tilastoinnin ja raportoinnin järjestelmien ylläpitäjät voitaisiin velvoittaa huolehtimaan selitteet paikoilleen kohtuullisen vähällä vaivalla. Kun saadaan luotua yhtenäinen tapa otsikoida asiat, se voidaan kirjata SAS-sovellusarkkitehtuuriin standardiksi.

Kehitettävä tietojärjestelmä tulee tarjoamaan uudenlaisen tavan tuottaa selitetekstejä Tietohakemiston tietokannasta perinteisen kovakoodaustyyllisen menettelyn sijaan. Kun eri ohjelmien käsittelemät selitetekstit tulevat yhdestä paikasta, selitteiden ylläpito helpottuu suuresti. Lisäksi tietoarkkitehtuurinäkökulmasta tietojen laatu tulee paranemaan, kun selitetekstien sisällöt ryhdytään tuottamaan tiedonhuoltoprosessin mukaisesti IT-osaston sijaan toimeksiantajaosastoilla, joilla on etuuksiin yms. liittyvää syvällistä substanssiosaamista.

Seuraavissa alakappaleissa esitellään yksityiskohtaisemmin sidosryhmän 3 odotukset ja tarpeet jaoteltuna tärkeisiin ja toissijaisiin tarpeisiin.

4.3.1 Ensisijaiset tarpeet

4.3.1.1 Tietojärjestelmän kehitystyö on toteutettava määriteltyjä standardeja noudattaen

Tietojärjestelmän kehitystyössä on noudatettava sovittuja standardeja; esimerkiksi ohjelmien koodauksessa on huomioitava ylläpidettävyyys. Myös ohjelmien sekä tiedostojen nimeämisessä on huomioitava standardit. Nimeämisissä on huomioitava esim. seuraavat seikat:

- Tietojärjestelmällä on systeemikoodi, joka ohjaa tiedostojen ja ohjelmien nimeämistä
- SQL-näkymät on nimettävä SQL-standardin mukaisesti
- Yhdistelytiedostot nimetään siten, että niillä on myös selitteet

4.3.1.2 Tietojärjestelmä on dokumentoitava hyvin

Jotta tietojärjestelmän jatkokehittäminen ja päivittäminen olisivat mahdollista, tietojärjestelmän kehitystyö ja tehdyt ratkaisut on dokumentoitava hyvin standardien mukaisesti. Tämä mahdollistaa myös ylläpitovastuiden siirtämisen henkilöltä toiselle.

4.3.1.3 Makron pitää tuottaa 1 määritellyllä hetkellä voimassa oleva selite-teksti

Tuotettuja selitetekstejä ei voi olla useampia kuin yksi, sillä kullakin tiedostolla / muuttujalla voi olla kerrallaan voimassa vain yksi kuvaus, josta teksti tuotetaan käyttäjälle. Käyttäjän tulee voida hakea vanhojen kuvausversioiden tekstejä aikaparametrin avulla.

4.3.1.4 Tietojärjestelmän tulee suorittaa lähdejärjestelmän purkuajot ajastetusti 1 krt / päivä

Purkuajojen tulee tapahtua päivittäin, jotta tilastoinnin ja raportoinnin tietojärjestelmien ylläpitäjät saavat mahdollisimman ajan tasaiset tiedot ohjelma- ja systeemitestausta varten.

4.3.1.5 Makron on oltava tarjolla sovitusta kirjastosta

Makro tulee viedä tuotantoympäristöön standardiohjelmistolla. Sen avulla versiointi tapahtuu automaattisesti sekä ajonohjauslauseet (JCL) generoituvat oikean muotoisina. Makron sijoituspaikka on mainittava myös ohjeistuksissa.

4.3.1.6 Makron käyttämisen on oltava yksinkertaista

Makron tulevien käyttäjien makro-osaaminen vaihtelee jonkin verran. Tästä johtuen makron käyttämisen tulee olla yksinkertaista, jotta se ei jää käyttämättä hankaluutensa vuoksi.

4.3.1.7 Makron tuleville käyttäjille on tiedotettava makrosta ja sen toiminnallisuudesta

Jotta mahdollisimman monet tilastointi- ja raportointijärjestelmien ylläpitäjät voisivat alkaa käyttää makroa, on sen käyttötapa ja toiminnallisuus selostettava ja ohjeistettava huolellisesti. Kohderyhmälle on tarpeen järjestää tiedotustilaisuus, jossa makron käyttöä myös demotaan.

4.3.1.8 Järjestelmää on voitava testata nykyhetkellä ja tulevaisuuteen; ettei tuota väärää tekstiä

Kuvaus on voitu määritellä siten, että se tulee voimaan vasta tietyssä ajankohtana. tietojärjestelmän on mahdollistettava myös tällaisten kuvausten tekstien testaaminen.

4.3.2 Toissijaiset tarpeet

4.3.2.1 Tietojärjestelmälle on määriteltävä tarkat käyttöoikeudet

Makron käyttö ei vaadi erityisiä käyttöoikeuksia, mutta tietojärjestelmän purkutiedostot sekä tekstitietojen koontitiedostot on suojattava siten, että niihin on vain lukuoikeudet (eivät ole käyttäjien muokattavissa). Selvitettävä vielä koneiden muokkausoikeudet.

5 Sidosryhmä 4; Tiedotettavat tahot

Tästä sidosryhmästä haastateltiin yhtä henkilöä. Hän edustaa yhteensä noin 5:tä sidosryhmäänsä kuuluvaa henkilöä.

5.1 Yleiskuvaus

Tähän ryhmään kuuluu tahoja, joiden on hyvä olla tietoisia tulevasta tietojärjestelmästä ja sen toiminnallisuudesta (esimerkiksi SAS-osaamisryhmä sekä TAKE-projekti). Haastateltavaksi valittiin nimenomaan SAS-osaamisryhmän edustaja, jonka tehtäviin kuuluu mm. SAS-arkkitehtuurin kehittäminen ja standardien luominen.

5.2 Yhteyshenkilöt

- IT-asiantuntija; SAS osaamisryhmän vetäjä

5.3 Tarpeet

Sidosryhmän edustaja toi esille, että kaikki sovitut standardit ja toimintatavat helpottavat toimimista isossa organisaatiossa. Esimerkiksi kehitettävä tietojärjestelmä voi tuoda selkeän ja yksiselitteisen tavan tuottaa SAS-tiedostojen ja -muuttujien selitteet hallitusti Tietohakemistosta.

Seuraavissa alakappaleissa esitellään yksityiskohtaisemmin sidosryhmän 4 odotukset ja tarpeet jaoteltuna tärkeisiin ja toissijaisiin tarpeisiin.

5.3.1 Ensisijaiset tarpeet

5.3.1.1 Kehitysprojekti on dokumentoitava sovitulla tavalla

SAS-osaamisryhmä pyrkii siihen, että heillä on tieto kaikista käytössä olevista SAS-makroista ja niiden toiminnallisuudesta. He toivovat, että kehitettävästä tietojärjestelmästä laaditaan dokumentaatio, josta selviää järjestelmän käyttötarkoitus, laajuus, käyttäjäryhmät sekä käyttötapaukset. He myös toivovat ajokaaviota, josta selviävät järjestelmään liittyvät ajot, ja luotavat tiedostot.

5.3.1.2 Tietojärjestelmään liittyvä SELITE-makro on sijoitettava standardeissa määriteltynä paikkaan standardiohjelmistoa käyttäen

Kaikki SAS-teknologiaan liittyvät makrot on pyritty sijoittamaan tiettyihin kirjastoihin, jotta ne löytyisivät helposti ja olisivat kaikkien niitä tarvitsevien käytettävissä. Myös SELITE-makro on sijoitettava samaan paikkaan. Lisäksi standardiohjelmiston hyödyntäminen on tarpeen sen vuoksi, että sen avulla tuotantoon vienti sujuu standardeja noudattaen, versiointi tapahtuu automaattisesti sekä JCL generoituu oikean muotoisena.

5.3.1.3 Tietojärjestelmän koodaus on suoritettava sovittujen standardien mukaisesti

Koska organisaatiossa on pyrkimys kehittää tieto- ja sovellusarkkitehtuuria nykyaikaiseen ja standardoituun suuntaan, on myös tämän uuden tietojärjestelmän koodaus suoritettava kaikkia määriteltyjä standardeja noudattaen. Kehitystyössä on hyvä ottaa myös kantaa siihen, mitä standardeja puuttuu eli millaisia linjauksia tarvittaisiin vielä jo olemassa olevien linjausten lisäksi.

5.3.1.4 Tietojärjestelmän käytöstä on järjestettävä tietoisuus

Tietoisuudessa tulee esitellä järjestelmän käyttämistä demon avulla. Olisi siis hyvä laatia joku konkreettinen käyttöesimerkki, jolla järjestelmää voidaan demota tuleville käyttäjille. Tietoisuudesta tulee informoida myös osaamisryhmää.

5.3.1.5 Tietojärjestelmälle on sovittava vastuuhenkilö

Uudelle järjestelmälle on nimettävä vastuuhenkilö. Vastuuhenkilön tehtävänä on säännöllisesti ja ainakin versionvaihtojen yhteydessä tarkistaa järjestelmän toimivuus. Vastuuhenkilö voi olla sellainen henkilö, joka on osallistunut järjestelmän rakentamiseen tai joka keskitetysti vastaa useammista järjestelmistä.

5.3.2 Toissijaiset tarpeet

5.3.2.1 Systeemikoodin valinta tulevalle tietojärjestelmälle

Kehitettävälle järjestelmälle on valittava sopiva systeemikoodi, joka vaikuttaa esim. ajojen ja tiedostojen nimeämisiin.

LIITE 1. Sidosryhmähaastatteluiden kysymykset

1. Sidosryhmän tai käyttäjän profiilin perustaminen

- a) Nimi
- b) Toimiala
- c) Työnimike
- d) Päävastuut
- e) Tuotettavat lopputuotteet / kenelle
- f) Miten onnistumista mitataan?
- g) Mitkä tekijät estävät onnistumista?
- h) Mitkä suuntaukset helpottavat / vaikeuttavat työtäsi?

2. Ongelman arviointi

- a) Miten tilastointitiedostojen, datalibien, muuttujien ja taulujen selitteet (=labelit) tuotetaan tällä hetkellä?
 - a. Mainitse esimerkki olemassa olevasta rakenteesta, jos ei hahmota kysymystä
 - b. Miten raportointijärjestelmien raporttien kuvaus esim. otsikko tuotetaan tällä hetkellä?
 - c. Miten tietuekuvaukset on tuotettu (esim. raporttien toimeksiantolomakkeet, ellei hahmota)?
- b) Missä ja miten selitteitä tällä hetkellä hyödynnetään?
- c) Mitä ongelmia nykykäytännöissä on ilmennyt? Miksi tämä on ongelma?
 - a. Mitä ongelmia selitteiden puuttuminen aiheuttaa?
- d) Miten toivoisit ongelmat ratkaistavan?

3. Pääkohtien kertaaminen ja analysoijan panos

- a) Listaa edustajan kuvaamat ongelmat
- b) Tuleeko muuta mieleen?
- c) Kerro, jos tiedät muita ongelmia, jotka saattavat liittyä tähän sidosryhmään?

4. Ratkaisun arviointi

- a) Esittele haastateltavalle uuden tietojärjestelmän ja SELITE-makron toimintaidea lyhyesti

5. Mahdollisuuksien arviointi

- a) Kuka organisaatiossa tarvitsee tätä sovellusta?
- b) Kuinka monta tämäntyyppistä käyttäjää käyttäisi sovellusta?
- c) Miten arvokkaana pitäisit onnistunutta ratkaisua?
- d) Missä ja miten tietohakemistoon tallennettuja kuvaustekstejä voisi mielestäsi hyödyntää ohjelmallisesti?

6. Käyttäjäympäristön ymmärtäminen

- a) Keitä käyttäjät ovat?
- b) Koulutustausta ja tietokoneosaaminen
- c) Ovatko tottuneita käyttämään tämänkaltaista sovellusta?
- d) Mitkä ohjelmisto/laitteistoalustat käytössä, entä tulevaisuudessa?
- e) Mihin muihin sovelluksiin tms. tarvitaan rajapinta?
- f) Mitä odotuksia käytettävyyden suhteen?
- g) Käyttöönotto; koulutuksen tarpeellisuus?

7. Luotettavuus-, suoritus- ja tukitarpeiden arviointi

- a) Mitä odotuksia luotettavuuden suhteen?
- b) Suorituskyvyn suhteen (esim. ajantasaisuus; kuinka usein purku)?
- c) Sovelluksen tuen järjestäminen? Erityistarpeita?
- d) Turvallisuusvaatimuksia?
- e) Asennus- ja konfiguraatiovaatimuksia? (esim. mitä muutoksia vaatii olemassa oleviin ohjelmiin)
- f) Lisensointivaatimuksia?
- g) Ohjelmiston jakelun hoitaminen?
- h) Nimeämisvaatimuksia?
- i) Muita vaatimuksia (mm. lainsäädäntö, standardit, ympäristövaatimukset jne.), mitä ohjelman kehittämisessä tulisi huomioida?

8. Jatkotoimenpiteet

- a) Muita kysymyksiä, jotka minun tulisi esittää?
- b) Voiko soittaa tai kysyä jatkokysymyksiä?
- c) Haluatko osallistua vaatimusten katselmointiin?

Liite 3. JAMES-lopputuote Visio

Visio

Tietohakemiston tekstejä hyödyntävä tietojärjestelmä

Vastuu: IT-Osasto /Järjestelmäkehitysryhmä / Miia Nopanen

Versionhallinta

Versio	Päiväys	Muokkaaaja	Kommentit
0.1	16.03.2010	Miia Nopanen	Aloitettu dokumentin laatiminen
0.2	03.04.2012	Miia Nopanen	Rakennekatselmoinnista saadut ehdotukset korjattu dokumenttiin
0.3	20.04.2012	Miia Nopanen	Työstämistä jatkettu
0.4	03.05.2012	Miia Nopanen	Työstämistä jatkettu
0.5	30.07.2012	Miia Nopanen	Työstämistä jatkettu, kappaleeseen 3.6 tuotettu tarpeiden prioriteetit
0.6	16.08.2012	Miia Nopanen	Hyväksytty versio luovutettavaksi seuraavan iteraation käyttöön

Dokumentin tiedot

Valmistumispäivä
Sijainti

17.08.2012

C:\Users\Miia\Desktop\Opinnäytetyö\Aloitusvaihe\Oparin liitteiksi\2_Visio v0.6.docx

Tulostettu
Mallipohjan versio

28.9.2012 22:26
JAMES 3.1

Sisällys

1 Johdanto	6
1.1 Tarkoitus	6
1.2 Laajuus	6
1.3 Määritelmät, akronyymit ja lyhenteet	6
1.4 Viittaukset.....	6
1.5 Yleiskuvaus.....	6
2 Ongelman kuvaus	7
3 Sidosryhmien ja käyttäjien kuvaukset.....	8
3.1 Yhteenveto sidosryhmistä	8
3.2 Yhteenveto käyttäjistä.....	9
3.3 Käyttäjän ympäristö.....	10
3.4 Sidosryhmien profiilit	10
3.4.1 Tilastollisen materiaalin loppukäyttäjät	10
3.4.2 SAS-sovellusarkkitehtuurin ja tietoarkkitehtuurin kehittäjät	11
3.4.3 Tiedotettavat tahot	12
3.5 Käyttäjien profiilit	13
3.5.1 SAS-teknologialla toteutettuja tilastointi- ja raportointijärjestelmiä kehittävät ja ylläpitävät henkilöt.....	13
3.5.2 Selainpohjaista raportointijärjestelmää kehittävät ja ylläpitävät henkilöt.....	14
3.6 Keskeiset sidosryhmien ja käyttäjien tarpeet	16
4 Tuotteen yleiskuvaus.....	22
4.1 Tuotenäkökulma	22
4.2 Ominaisuuksien yhteenveto.....	23

4.3 Oletukset ja riippuvuudet.....	23
5 Tuotteen ominaisuudet.....	24
5.1 Tietojärjestelmän käytön on oltava yksinkertaista	24
5.2 Tietojärjestelmän on toimittava määritysten mukaisesti oikein	24
5.3 Tietojärjestelmän kehityksessä on käytettävä standardiohjelmistoa, ja makro on tallennettava sovittuun paikkaan	24
5.4 Tietohakemiston tietokannan purun on tapahduttava säännöllisesti ja oikein ajoitettusti, ja purku on voitava suorittaa myös tarvittaessa.....	24
5.5 Pystyy tuottamaan eri kuvausversioiden selitetekstejä.....	24
5.6 Toiminta virhetilanteissa on suunniteltava huolellisesti.....	24
5.7 Makrokutsun koodaamisen muihin tietojärjestelmiin on oltava helppoa ..	24
5.8 Tietojärjestelmän ylläpito on tehtävä vaivattomaksi.....	25
5.9 Tietojärjestelmää hyödynnetään muissa tietojärjestelmissä SELITE-makron avulla.....	25
6 Rajoitukset.....	26
7 Laatumittarit.....	26
8 Tärkeys ja prioriteetti	26
9 Muut tuotteen vaatimukset	27
9.1 Sovellettavat standardit	27
9.2 Järjestelmän vaatimukset.....	27
9.3 Suorituskykyvaatimukset.....	27
9.4 Ympäristövaatimukset.....	27
10 Dokumentoinnin vaatimukset.....	28
10.1 Asiakkaan ohjeistus	28
11 A Ominaisuuksien ominaisuudet.....	29
11.1 A.1 Tila	29
11.2 A.2 Hyöty	29

11.3 A.3 Työmäärä	29
11.4 A.4 Riski	30
11.5 A.5 Vakaus	30
11.6 A.7 Vastuuhenkilö	30
11.7 A.8 Prioriteetti	30

1 Johdanto

1.1 Tarkoitus

Visio-dokumentin tarkoituksena on varmistaa, että kaikilla sidosryhmillä ja projektin jäsenillä on sama näkemys siitä, mitä kehitetään ja miksi. Tähän dokumenttiin kerätään, analysoidaan ja määritellään Tietohakemiston tekstejä hyödyntävän tietojärjestelmän korkean tason tarpeet ja ominaisuudet. Dokumentti keskittyy sidosryhmien ja tulevien käyttäjien tarvitsemiin ominaisuuksiin ja siihen, miksi heillä on näitä tarpeita. Yksityiskohdat, millä tavoin Tietohakemiston tekstejä hyödyntävä tietojärjestelmä täyttää nämä tarpeet, tullaan kuvaamaan muissa dokumenteissa.

1.2 Laajuus

Tämä dokumentti liittyy Tietohakemiston tekstejä hyödyntävien tietojärjestelmien kehitysprojektiin. Se kuuluu osana Kelassa laajemmin meneillään olevaan tietoarkkitehtuurin kehittämiseen, jonka tavoitteena on kehittää tiedonhuoltoprosessia ja parantaa organisaatiossa käytettävien tietojen laatua.

Tämä dokumentti toimii lähtöaineistona seuraaville projektissa myöhemmin tuotettaville dokumenteille:

- Käyttötapausmalli (kokoava)
- Systeemitestisuunnitelma.

1.3 Määritelmät, akronyymit ja lyhenteet

Dokumentin kannalta olennaiset termit ja lyhenteet on kirjattu Sanasto-dokumenttiin, jota päivitetään koko projektin ajan (ks. kappale 1.4).

1.4 Viittaukset

Tämän dokumentin lähdeaineistona on käytetty seuraavia dokumentteja:

- Sidosryhmien tarpeet -dokumentti, johon on koottu neljän tärkeimmän sidosryhmän toiveet ja tarpeet kehitettävälle järjestelmälle. Sidosryhmän tarpeet -dokumentti löytyy verkkolevyltä projektin kansiota.
- Sanasto-dokumentti, johon on koottu projektin ja kehitettävän tietojärjestelmän kannalta olennaiset termit ja määritelmät. Sanasto-dokumenttia päivitetään koko projektin ajan. Dokumentti löytyy verkkolevyltä projektin kansiota.

1.5 Yleiskuvaus

Dokumentissa esitellään ensin ongelma, jonka tuotettava tietojärjestelmä tulee ratkaistaan. Seuraavaksi esitellään sidosryhmät ja käyttäjät, joihin kehitystyö ja sen tuloksena oleva järjestelmä tulee vaikuttamaan. Näiden ryhmien järjestelmälle esittämät tarpeet on taulukoitu ja priorisoitu kappaleessa 3.6. Kappaleessa 4 luodaan kehitettävästä tietojärjestelmästä yleiskuvaus, ja kappaleeseen 10 on koottu järjestelmän ominaisuuksia. Kappale 11 (Ominaisuuksien ominaisuudet) on vielä keskeneräinen, ja sitä tullaan täydentämään myöhemmissä iteraatioissa.

2 Ongelman kuvaus

Ongelma	SAS-tiedostojen ja niissä olevien muuttujien selitteet eli labelit tuotetaan usein IT-osaston henkilön toimesta kirjoittamalla ne muun koodin sisälle tai ohjelmallisesti tietystä SAS-jäsenestä, sillä muuta keinoa ei ole olemassa. Käytäntö tuottaa ongelmia sekä järjestelmien ylläpitäjille että loppukäyttäjille.
vaikuttaa	<ul style="list-style-type: none"> SAS-teknologialla toteutettuja tilastointi- ja raportointijärjestelmiä kehittävät ja ylläpitävät henkilöt Tilastollisen materiaalin loppukäyttäjät
minkä seurauksena	<ul style="list-style-type: none"> Selitteiden määrittely ja kirjaaminen tapahtuvat usein IT-henkilöiden toimesta. Selitteiden ylläpito on työlästä, ja selitteet jäävät joskus kokonaan kirjaamatta. Tästä johtuen SAS-tiedostojen ja -muuttujien selitteet ovat joskus epäselviä tai virheellisiä, tai ne puuttuvat kokonaan. Loppukäyttäjille virheelliset tai puutteelliset selitteet voivat aiheuttaa väärinymmärryksiä ja niiden selvittäminen vie aikaa
onnistunut ratkaisu olisi	<p>Selitteet tuotettaisiin uuden tietojärjestelmän avulla Tietohakemistosta ohjelmallisesti. Tästä olisi seuraavia hyötyjä:</p> <ul style="list-style-type: none"> Selitteiden sisällöt olisivat substanssiosaimista omaavien henkilöiden määrittämiä Selitteet tulisivat ohjelmallisesti Tietohakemistosta, jossa ne ovat oikeita ja varmasti ajan tasalla Tietojärjestelmien ylläpito helpottuu

3 Sidosryhmien ja käyttäjien kuvaukset

3.1 Yhteenveto sidosryhmistä

Tässä esitellään projektin sidosryhmät taulukon muodossa. Sidosryhmät on kuvattu myös Sidosryhmien tarpeet -dokumentissa

Nimi	Kuvaus	Vastuut
Tilastollisen materiaalin loppukäyttäjät	<p>Tämän sidosryhmän henkilöt työskentelevät pääsääntöisesti muilla osastoilla kuin IT-osastolla. He toimivat usein toimeksiantajan roolissa, eli he tilaavat IT-osastolta tutkimuskäyttöön tarvittavia tilastointitiedostoja.</p> <p>Tilastointitiedostojen ja niissä olevien muuttujien selitteiden perusteella sidosryhmän henkilöt saavat tärkeää tiedoston ja muuttujien sisällöstä ja tarkoituksesta.</p>	Sidosryhmä ei itse käytä tietojärjestelmää, mutta se tulee hyötymään tietojärjestelmän toiminnallisuuden tuloksista eli oikeista ja ajan tasaisista selitteistä.
SAS-sovellusarkkitehtuurin ja tietoarkkitehtuurin kehittäjät	Tähän sidosryhmään kuuluu henkilöitä ja tahoja, jotka pyrkivät tuomaan esille ja kehittämään SAS-sovellusarkkitehtuuriin sekä tietoarkkitehtuuriin liittyviä asioita.	<p>Sidosryhmä eivät itse tule varsinaisesti käyttämään tietojärjestelmää tai makroa, mutta heillä on omia arkkitehtuurien kehittämiseen liittyviä odotuksia ja tarpeita järjestelmälle.</p> <p>Lisäksi tähän sidosryhmään kuuluu projektin omistaja, joka seuraa projektin etenemistä.</p>

Tiedotettavat tahot	Tähän ryhmään kuuluu tahoja, joiden on hyvä olla tietoisia tulevasta tietojärjestelmästä ja sen toiminnallisuudesta	<p>Tähän sidosryhmään kuuluva SAS-osaamisryhmä koordinoi SAS-teknologiaan liittyviä asioita ja edistää SAS-osaamista ja sovellusarkkitehtuuria organisaatiossa. He tulevat myös tarjoamaan tukea tietojärjestelmän rakennusvaiheessa sekä ottamaan kantaa tietojärjestelmän ylläpitoon tulevaisuudessa esimerkiksi versionvaihtojen yhteydessä.</p> <p>Muut tiedotettavat tahot ovat projekteja tai organisaatiotahoja, joiden on hyvä olla tietoisia kehitettävästä tietojärjestelmästä ja sen tarjoamista mahdollisuuksista.</p>
---------------------	---	--

3.2 Yhteenveto käyttäjistä

Seuraavassa taulukossa esitellään projektin tuotoksen käyttäjät. Lisätietoa ryhmästä löytyy Sidosryhmien tarpeet -dokumentista.

Nimi	Kuvaus	Vastuut	Sidosryhmä
SAS-teknologialla toteutettuja tilastointi- ja raportointijärjestelmiä kehittävät ja ylläpitävät henkilöt	Tämän sidosryhmän edustajat työskentelevät IT-osastolla. Sidosryhmään kuuluvat henkilöt ovat sellaisia, jotka tulevat nimenomaan käyttämään tulevaa tietojärjestelmää ja SELITE-makroa omissa ohjelmissaan.	Heidän työtehtäviinsä kuuluu mm. SAS-teknologialla toteutettujen tilastointi- ja raportointijärjestelmien kehitys ja ylläpito.	

3.3 Käyttäjän ympäristö

Tietojärjestelmän tulevat käyttäjät työskentelevät pääosin omilla työpisteillään, ja työtehtävät suoritetaan tällä hetkellä yleensä pääkoneympäristössä. Tulevaisuudessa työskentely siirtyy kuitenkin enemmän palvelin- ja työasemaympäristöön, ja työ voi olla liikkuvaakin, esimerkiksi etätöitä. Työasemaohjelmistoista erityisesti SAS/EG ja MikroSAS tulevat yleistymään, joten tietojärjestelmän tulee tukea myös niitä.

Yhtä tilastointi- ja raportointijärjestelmää kehittää ja ylläpitää yleensä yksi henkilö, jonka vastuulle järjestelmä on nimetty. Ajoittain (esimerkiksi lomien aikana) myös muut henkilöt voivat joutua käsittelemään järjestelmää.

Kehitettävää tietojärjestelmää tullaan käyttämään osana jo olemassa olevia tietojärjestelmiä.

3.4 Sidosryhmien profiilit

Tässä kappaleessa esitellään ne projektin sidosryhmät, jotka eivät varsinaisesti tule itse käyttämään tulevaa tietojärjestelmää. Järjestelmän loppukäyttäjät kuvataan kappaleessa 3.5.

3.4.1 Tilastollisen materiaalin loppukäyttäjät

Edustaja	Aktuaariasiantuntija / IT-osaston sisäinen asiakasosasto Tutkija / IT-osaston sisäinen asiakasosasto
Kuvaus	Tämän sidosryhmän jäsenet työskentelevät muilla kuin IT-osastolla. Sidosryhmän jäsenten toimenkuvaan kuuluu erilaisten tilastojen ja tutkimusten laatiminen tilastollisen taustamateriaalin pohjalta. Sidosryhmän jäsenet toimivat usein toimeksiantajan roolissa, eli he tilaavat IT-osastolta tutkimuskäyttöön tarvittavia tilastointitiedostoja.
Tyyppi	Sidosryhmä ei itse käytä tietojärjestelmää, mutta se tulee hyötymään tietojärjestelmän toiminnallisuuden tuloksista eli oikeista ja ajan tasaisista selitteistä.
Vastuut	Toimiva tietojärjestelmä mahdollistaa oikeiden ja ajan tasalla olevien selitteiden tuottamisen tilastointitiedostoihin ja niissä oleviin muuttujiin, mikä helpottaa sidosryhmän jäsenten työtä huomattavasti.
Onnistumiskriteerit	Tärkeimpiä onnistumiskriteerejä ovat virheettömät ja oikein tuotetut datat, tilastot ja tutkimukset sekä annetussa aikataulussa pysyminen. Erityisesti lainmuutoksiin liittyvissä tehtävissä aikataulu on usein tiukka, ja siinä pysyminen erityisen tärkeää.
Osallistuminen	Sidosryhmä ei varsinaisesti osallistu projektiin, mutta hyöttyy projektin aikaansaannoksista.

Tuotokset	Sidosryhmä toivoo, että heille tiedotetaan tietojärjestelmän käyttöönoton yhteydessä tietojärjestelmän toiminnallisuuksista ja käytöstä.
Kommentit / kysymykset	Sidosryhmä pitää kehitystyötä tärkeänä, sillä oikeat ja ajan tasalla olevat selitteet helpottavat ja nopeuttavat työskentelyä huomattavasti.

3.4.2 SAS-sovellusarkkitehtuurin ja tietoarkkitehtuurin kehittäjät

Edustaja	Sovelluskehittäjien esimies / IT-osasto
Kuvaus	Sidosryhmän jäsenten toimenkuvaan kuuluu SAS-teknologiaan liittyvän sovellusarkkitehtuurin sekä tietoarkkitehtuurin kehittäminen ja esille tuominen. Ryhmän edustajan tehtäviin kuuluu myös kehitettävään tietojärjestelmään liittyvän Tietohakemiston toimintaa ja laatua kuvaavien raporttien tuottaminen.
Tyyppi	Sidosryhmä eivät itse tule varsinaisesti käyttämään tietojärjestelmää tai makroa, mutta heillä on omia arkkitehtuurien kehittämiseen liittyviä odotuksia ja tarpeita järjestelmälle.
Vastuut	Sidosryhmä edustaja vastaa SAS-teknologiaan liittyvän sovellusarkkitehtuurin kehittämisestä ja antaa myös linjauksia kehitettävälle tietojärjestelmälle. Lisäksi tähän sidosryhmään kuuluu projektin omistaja, joka seuraa projektin etenemistä.
Onnistumiskriteerit	SAS-sovellusarkkitehtuuria saadaan kehitettyä eteenpäin esimerkiksi kuvaamalla ja määrittelemällä siihen liittyviä asioita. Kehitettävä tietojärjestelmä toimii eräänlaisena onnistumiskriteerinä, sillä sen puitteissa voidaan tutkia ja laatia erilaisia standardeja vastaavanlaisten järjestelmien kehittämistä varten. Onnistumiskriteeri on myös toimiva raportti, jolla voidaan tuottaa informaatiota projekteille tietokuvausten käsittelytilanteesta Tietohakemistossa.

Osallistuminen	<p>Sidosryhmä katselmoi haastatteluissa tuottamansa tarpeet ja tarkentaa niitä tarvittaessa.</p> <p>Sidosryhmä seuraa projektin etenemistä ja laatii määräytyksiä ja standardeja tietojärjestelmän kehitystyölle.</p> <p>Lisäksi he tulevat opastamaan SAS-teknologiaa ja organisaation toimintamalleja niiltä osin, kuin niitä on tarpeen tuntea tietojärjestelmän kehittämiseksi.</p>
Tuotokset	Sidosryhmä edellyttää, että tietojärjestelmän kehitystyö tehdään JAMES-menetelmän mukaisesti ja että siitä tuotetaan sovitut JAMES-dokumentit.
Kommentit / kysymykset	Selitteiden tuottaminen ja käyttö on saatava kirjattua SAS-sovellusarkkitehtuuriin. Lisäksi kehitystyössä on noudatettava sovittuja standardeja ja määräytyksiä.

3.4.3 Tiedotettavat tahot

Edustaja	IT-asiantuntija ja osaamisryhmän edustaja / IT-osasto
Kuvaus	<p>Tähän sidosryhmään kuuluu tahoja, joiden on hyvä olla tietoisia kehitettävästä tietojärjestelmästä ja sen toiminnallisuudesta.</p> <p>Haastateltavaksi valittiin nimenomaan SAS-osaamisryhmän edustaja, jonka tehtäviin kuuluu mm. SAS-teknologiaan liittyvien asioiden koordinoiminen ja niistä tiedottaminen organisaatiossa. Lisäksi osaamisryhmä osallistuu SAS-sovellusarkkitehtuurin kehittämiseen ja standardien luomiseen.</p> <p>Muut tiedotettavat tahot ovat projekteja tai organisaatiotahoja, joiden on hyvä olla tietoisia kehitettävästä tietojärjestelmästä ja sen tarjoamista mahdollisuuksista. He saavat tietoa joko SAS-osaamisryhmän tai projektin välityksellä.</p>
Tyyppi	Sidosryhmä eivät itse tule varsinaisesti käyttämään tietojärjestelmää tai makroa, mutta heillä on omia arkkitehtuurien kehittämiseen tai toiminnan organisointiin liittyviä odotuksia ja tarpeita järjestelmälle.
Vastuut	Sidosryhmä tulee tarjoamaan teknistä tukea tietojärjestelmän rakennusvaiheessa sekä ottamaan kantaa tietojärjestelmän ylläpitoon tulevaisuudessa esimerkiksi versionvaihtojen yhteydessä.

Onnistumiskriteerit	SAS-teknologiaan liittyvien standardien luominen ja kehittäminen; työskentelyn määrämuotoistaminen ja standardoiminen helpottavat ja nopeuttavat työskentelyä
Osallistuminen	<p>Sidosryhmä katselee haastatteluissa tuottamansa tarpeet ja tarkentaa niitä tarvittaessa.</p> <p>Sidosryhmä seuraa projektin etenemistä ja laatii määräytyksiä ja standardeja tietojärjestelmän kehitystyölle.</p> <p>Lisäksi he tulevat opastamaan SAS-teknologiaa ja organisaation toimintamalleja niiltä osin, kuin niitä on tarpeen tuntea tietojärjestelmän kehittämiseksi.</p>
Tuotokset	<p>Sidosryhmä toivoo, että projekti toimittaa kehitettävästä tietojärjestelmästä kattavan dokumentaation, joka sisältää mm.</p> <ul style="list-style-type: none"> * Järjestelmän käyttötarkoituksen ja laajuuden * Käyttäjäryhmät ja käyttötapaukset * Ajokaavion sisältäen ajot ja luotavat tiedostot
Kommentit / kysymykset	SAS-sovellusarkkitehtuurin ja sen sisältämien standardien laatiminen ovat vielä hyvin alkuvaiheessa. Tästä johtuen kehitettävälle tietojärjestelmälle ei välttämättä ehditä laatia kaikkia määräytyksiä projektin puitteissa.

3.5 Käyttäjien profiilit

3.5.1 SAS-teknologialla toteutettuja tilastointi- ja raportointijärjestelmiä kehittävät ja ylläpitävät henkilöt

Tämä käyttäjäryhmä on tietojärjestelmän pääasiallinen käyttäjäryhmä.

Edustaja	<p>Esimies / IT-osasto</p> <p>IT-asiantuntija /IT-osasto</p>
Kuvaus	Tämän käyttäjäryhmän henkilöt kehittävät ja ylläpitävät nimenomaan SAS-teknologialla toteutettuja tilastointi- ja raportointijärjestelmiä.
Tyyppi	<p>Käyttäjäryhmän jäsenillä on pääsääntöisesti vahvaa osaamista tilastointi- ja raportointijärjestelmien ylläpidosta pääkoneympäristössä, mutta työasemaohjelmistojen kanssa toimimista ollaan vasta aloittelemassa.</p> <p>Käyttäjäryhmään voi kuitenkin tulevaisuudessa kuulua myös sellaisia henkilöitä, joilla ei ole lainkaan kokemusta Kelan tietojärjestelmistä, SAS-teknologiasta tai pääkoneympäristöstä.</p>

Vastuut	<p>Tämän käyttäjäryhmän henkilöt tulevat käyttämään uutta tietojärjestelmää ja lisäävät makrokutsun omiin ohjelmiinsa.</p> <p>Lisäksi järjestelmän ylläpitovastuu siirtyy mahdollisesti jollekin tämän ryhmän henkilölle.</p>
Onnistumiskriteerit	Omien vastuualueiden ohjelmat toimivat saumattomasti yhteen uuden tietojärjestelmän kanssa, ajot eivät kaadu tms.
Osallistuminen	<p>Käyttäjäryhmä katselmoi haastatteluissa tuottamansa tarpeet ja tarkentaa niitä tarvittaessa.</p> <p>Osa ryhmän jäsenistä osallistuu uuden tietojärjestelmän testaamiseen.</p> <p>Lisäksi ryhmän käyttäjät osallistuvat tietojärjestelmän käyttöönottoon siten, että he tekevät tarvittavat muutokset omiin tilastointi- ja raportointijärjestelmiinsä.</p>
Tuotokset	Käyttäjät esittivät omat tarpeensa kehitettävälle järjestelmälle. Tarpeet selvitettiin haastatteluilla, ja ne kirjattiin Sidosryhmien tarpeet -dokumenttiin.
Kommentit / kysymykset	<p>Projektissa voi tuottaa vaikeuksia:</p> <ul style="list-style-type: none"> * Organisaation suuri koko ja tietynlainen kankeus uudistuksissa * Käyttäjien sitoutumattomuus uuteen tapaan tehdä työtä uuden tietojärjestelmän avulla * Käyttäjien makro-osaamisen vaihteleva taso * Mahdolliset ongelmat tietojärjestelmän tai makron toiminnassa; vievät uskottavuutta ja hankaloittavat työskentelyä <p>Projektia voi helpottaa:</p> <ul style="list-style-type: none"> * Organisaation yleinen pyrkimys työskentelyn standardointiin ja määrämuotoistamiseen * Hyvä tiedottaminen ja ohjeistaminen sekä kehitystyön aikana että käyttöönoton yhteydessä * Toimiva tietojärjestelmä helpottaa työskentelyä

3.5.2 Selainpohjaista raportointijärjestelmää kehittävät ja ylläpitävät henkilöt

Tämä käyttäjäryhmä ei ole varsinaisesti projektin pääasiallinen käyttäjäryhmä, sillä projektissa tähdätään aluksi SAS-teknologialla toteutettujen järjestelmien ongelmien ratkaisemiseen. Selainpohjaisen raportointijärjestelmän kannalta ongelmat eivät ole tällä hetkellä niin akuutteja, ja sen vuoksi järjestelmän tarpeita palvelevat toiminnallisuudet tullaan kehittämään tämän projektin ulkopuolella.

Käyttäjärühmä kuitenkin huomioidaan tässä projektissa mahdollisena tulevana käyttäjäryhmänä.

Edustaja	Sovelluskehittäjien esimies /IT-osasto
Kuvaus	Tähän käyttäjäryhmään kuuluvat henkilöt kehittävät ja ylläpitävät Kelan dynaamista selainpohjaista raportointijärjestelmää.
Tyyppi	Käyttäjärühmän jäsenillä on pääsääntöisesti vahvaa osaamista raportointijärjestelmän ylläpidosta WebFOCUS-tekniikalla. Käyttäjärühmään voi kuitenkin tulevaisuudessa kuulua myös sellaisia henkilöitä, joilla ei ole lainkaan kokemusta Kelan tietojärjestelmistä tai WebFOCUS-tekniikasta.
Vastuut	Käyttäjärühmän henkilöt tulevat antamaan perustiedot ja määrittäimet, millaisia toiminnallisuuksia selainpohjainen raportointijärjestelmä tulee tarvitsemaan kehitettävältä järjestelmältä. Lisäksi he tulevat opastamaan WebFOCUS-tekniikkaa niiltä osin, kuin sitä on tarpeen tuntea tietojärjestelmän kehittämiseksi.
Onnistumiskriteerit	Toimivan, määrittämisen täyttävän ja toimeksiantajan hyväksymän raportin tuottaminen tuotantoon annetussa aikataulussa. Käyttöönoton jälkeen ilmenevien virhetilanteiden vähäinen määrä.
Osallistuminen	Käyttäjärühmä katselmoi haastatteluissa tuottamansa tarpeet ja tarkentaa niitä tarvittaessa.
Tuotokset	Tässä vaiheessa käyttäjäryhmä ei tuota mitään.
Kommentit / kysymykset	Projektissa voi tuottaa vaikeuksia: <ul style="list-style-type: none"> * Käyttäjien sitoutumattomuus uuden tietojärjestelmän käyttämiseen * Ennalta arvaamattomat muutostarpeet * Tekniset ongelmat * Erinimiset muuttujat (samalla tietosisällöllä) * Samannimiset muuttujat eri raporteilla; muutokset näkyvät kaikilla raporteilla Projektia voi helpottaa: <ul style="list-style-type: none"> * Organisaation yleinen pyrkimys työskentelyn standardointiin ja määrämuotoistamiseen * Selainpohjaisen raportointijärjestelmän ylläpitäjät ovat tottuneet yleisöiden käyttämiseen omissa raporttiohjelmissaan

3.6 Keskeiset sidosryhmien ja käyttäjien tarpeet

Seuraavaan taulukkoon on listattu kaikki Sidosryhmien tarpeet -dokumenttiin kootut, haastatteluissa esiin nousseet tarpeet. Huolenaihe-sarakkeeseen on kirjattu seikkoja, jotka voivat vaikeuttaa työskentelyä tai tuottaa ongelmia, mikäli siihen liittyvää tarvetta ei huomioida järjestelmän kehitystyössä. Nykyinen ratkaisu -sarakkeeseen on kirjattu, miten asia on tällä hetkellä huomioitu vastaavanlaisissa tilanteissa - jos on huomioita. Ehdotettu ratkaisu -sarakkeeseen on kirjattu, kuinka kyseinen tarve voitaisiin ottaa huomioon kehitystyössä.

Prioriteetti-sarakkeeseen on merkitty kunkin tarpeen prioriteetit kehitystyössä. Arvot ovat välillä 1-3 siten, että 1= tarve tulee huomioida kehitystyössä, 2=hyvä huomioida 3=voidaan toteuttaa myöhemmissä versioissa. Prioriteettien tuottaminen suoritettiin siten, että osa sidosryhmien edustajista (2 hlöä) antoi itsenäisesti tarpeille prioriteetteja väliltä 1-3. Kutakin prioriteettia piti käyttää vähintään 4 kertaa, jotta esim. kaikille tarpeille ei voinut antaa arvoa 1. Näin saatiin aikaan eroja tarpeiden välille. Lopuksi annetuista prioriteeteista laskettiin keskiarvo, joka kirjattiin tarpeen lopulliseksi prioriteetiksi. Tarpeet on järjestetty prioriteetin mukaisesti laskevaan järjestykseen.

Tarve	Prioriteetti	Huolenaihe	Nykyinen ratkaisu	Ehdotettu ratkaisu
Tietojärjestelmän on toimittava määritysten mukaisesti oikein; tuotettava vain 1 määritellyllä hetkellä voimassa oleva selite-teksti	1	Tietojärjestelmän toiminnassa on virheitä, eikä siihen voi luottaa	Olemassa olevaa ratkaisua ei ole	Tietojärjestelmän toimintaa testataan ennen käyttöönottoa sekä toteuttajien että käyttäjien toimesta huolellisesti, jotta mahdolliset virheet saadaan korjattua

Tietojärjestelmän ja makron käytön on oltava yksinkertaista	1,5	Tietojärjestelmä ja siihen liittyvä makro ovat niin hankalia käyttää, että käyttäjät eivät halua hyödyntää niiden toiminnallisuksia	Tilastointi- ja raportointijärjestelmissä hyödynnettävät makrot ovat toiminnaltaan hyvin erilaisia. Osa vaatii syvempää ymmärrystä, osan käyttö on yksinkertaista.	Pyritään huomioidmaan käyttäjärühmien tarpeet ja toiveet mahdollisimman hyvin. Järjestelmä ja makro katselmoidaan ja testataan tulevien käyttäjien toimesta, ja tarpeen mukaan niiden toimintaa muokataan saadun palautteen myötä käyttäjäystävällisemmiksi
Tietojärjestelmän purkuajot on ajastettava säännöllisiksi (1krt/pvä-1krt/kk), mutta purun on onnistuttava myös tarpeen vaatiessa	1,5	Purkuajot on ajastettu liian harvoiksi, jolloin järjestelmä ei tuota ajan tasalla olevia tekstejä	Muiden tietojärjestelmien purkuajot on ajastettu kutakin järjestelmää parhaiten palvelevasti. Lisäksi niitä voi purkaa tarpeen mukaan	Määritellään, mikä olisi sopiva purkutiheys ja ajastetaan purkuajot sen mukaisesti huomioiden pääkoneen ruuhkahuiput. Lisätään järjestelmään myös mahdollisuus suorittaa purkuajo tarpeen vaatiessa.

Tietojärjestelmän käyttönotosta ja sen toiminnasta on laadittava selkeä ohjeistus	1,5	Tietojärjestelmän käyttäjät eivät osaa käyttää järjestelmää, jolloin sitä ei pystytä hyödyntämään toivotulla tavalla	Tietojärjestelmistä laadittuja ohjeita ei ole standardoitu, ja niitä on tuotettu tietojärjestelmäkehityksessä hyvin eritavoin ja eri laajuudella	Laaditaan selkeä ohjeistus, jonka avulla tietojärjestelmiä tai SAS-teknologiaa tuntemattomien henkilöiden pystyvät käyttämään uutta tietojärjestelmää
Tietojärjestelmän tulee voida tuottaa Tietohakemistosta myös vanhempien kuvausversioiden tietoja aikaparametrilla	1,5	Tietojärjestelmä pystyy tuottamaan vain viimeisimmän kuvauksen tekstejä, vaikka usein tarvitaan myös tietyn ajan kohdan tekstejä vanhemmista kuvauksista	Olemassa olevaa ratkaisua ei ole	Mahdollistetaan eri kuvausversioiden tekstien tuottaminen aikaparametrin avulla
Tietojärjestelmästä ja sen toiminnallisuudesta on tiedotettava kaikkia sidos- ja käyttäjäryhmiä	2	Tietojärjestelmän käyttäjät tai siitä muuten hyötyvät tahot eivät ole tietoisia järjestelmästä ja sen toiminnallisuudesta, jolloin niitä ei osata hyödyntää	Uusien tietojärjestelmien käytöstä on tiedotettu olettettuja käyttäjiä lähinnä omien tiimien sisällä melko suppeasti	Järjestetään lyhyitä tietokäyttösuureksia, joissa uutta tietojärjestelmää esitellään ja sen käyttöä demotaan hyvien esimerkkien avulla

Tietojärjestelmälle on nimettävä ylläpitäjä ja tietojärjestelmä on suojattava tahattomalta muokkaukselta	2	Tietojärjestelmää tai siihen liittyvää makroa ei saa päästä muokkaamaan tarpeettomasti	Olemassa oleville tietojärjestelmille on nimetty vastuuhenkilöt, jotka huolehtivat esim. järjestelmien päivittämisestä	Muokkausoikeudet vain tietojärjestelmän ylläpitäjälle
Makroa kutsuva ohjelma ei saa kaatua, vaikka makro ei onnistuisikaan tuottamaan kaikkia käyttäjän kutsuun määrittelemiä selitteitä	2	Tärkeät tilastointi- ja raportointijärjestelmät kaatuvat makron toimimattomuuden vuoksi	Ajonaikaisista virheistä lähtee tieto ryhmän sähköpostilaatikkoon, ja järjestelmän vastuuhenkilö käy korjaamassa tilanteen niin pian kuin mahdollista	Rakennetaan uusi tietojärjestelmä sellaiseksi, että se tuottaa lopputuloksen makron toimintavirheestä huolimatta. Selvitetään, voisiko IT-palveluiden hallintajärjestelmää hyödyntää virhetilanteista tiedottamiseen.
Tietojärjestelmän kehitystyö on toteutettava määriteltyjä standardeja noudattaen	2	Kehitystyössä ei noudateta standardeja, jolloin ylläpito ja jatkokehitys vaikeutuvat. Lisäksi muiden on vaikeaa todentaa ja hyödyntää tietojärjestelmän toiminnallisuksia	Olemassa olevien tietojärjestelmien kehitystyössä on noudatettu standardeja vaihtelevasti. Pahimmillaan sovittuja standardeja ei ole noudatettu eikä minkäänlaista dokumentaatiota laadittu.	SAS-sovel-lusarkkitehtuuria ja SAS-osaamis-ryhmää edustavat henkilöt ilmoittavat kehitystyössä noudatettavat standardit

Tietojärjestelmä on dokumentoitava määritellyllä tavalla	2	Tietojärjestelmästä ei laadita dokumentaatiota, jolloin kaikki tietojärjestelmään liittyvä tieto on ainoastaan laatijansa tiedossa. Tästä voi aiheutua suuria ongelmia muutostilanteissa ja ylläpitovastuiden siirtäessä muille henkilöille	Tietojärjestelmien dokumentointi on melko kirjavaa ja satunnaista. Yhteisiä standardeja tuotettavista dokumenteista ei ole laadittu.	Kehitystyötä tuotetaan seuraavat dokumentit: * JAMES-menettelmään liittyvät sovitut dokumentit * Järjestelmän käyttötarkoituksen ja laajuuden kuvaavat dokumentit * Käyttäjärhyvät ja käyttötapaukset * Ajokaavion sisältäen ajot ja luotavat tiedot * Lisäksi koodia kommentoidaan riittävästi
Makro on toteutettava siten, ettei käyttäjän tarvitse kirjoittaa makrokutsuun parametreiksi kaikkia teknisiä nimiä erikseen	2,5	Käyttäjä joutuu kirjoittamaan makrokutsuun tekniset nimet erikseen, eikä työ määrä pie-nene verrattuna nykyiseen. Tämä ei motivoi siirtymään uuden tietojärjestelmän käyttäjäksi	Olemassa olevien makrojen toiminnallisuudet vaihtelevat; joihinkin käyttäjä joutuu kirjaamaan parametreja itse, joihinkin niitä voidaan tuottaa ohjelmallisesti	Makro tuottaa muuttujen tekniset nimet SAS-ohjelmointikielen proc contents-proseduurin avulla

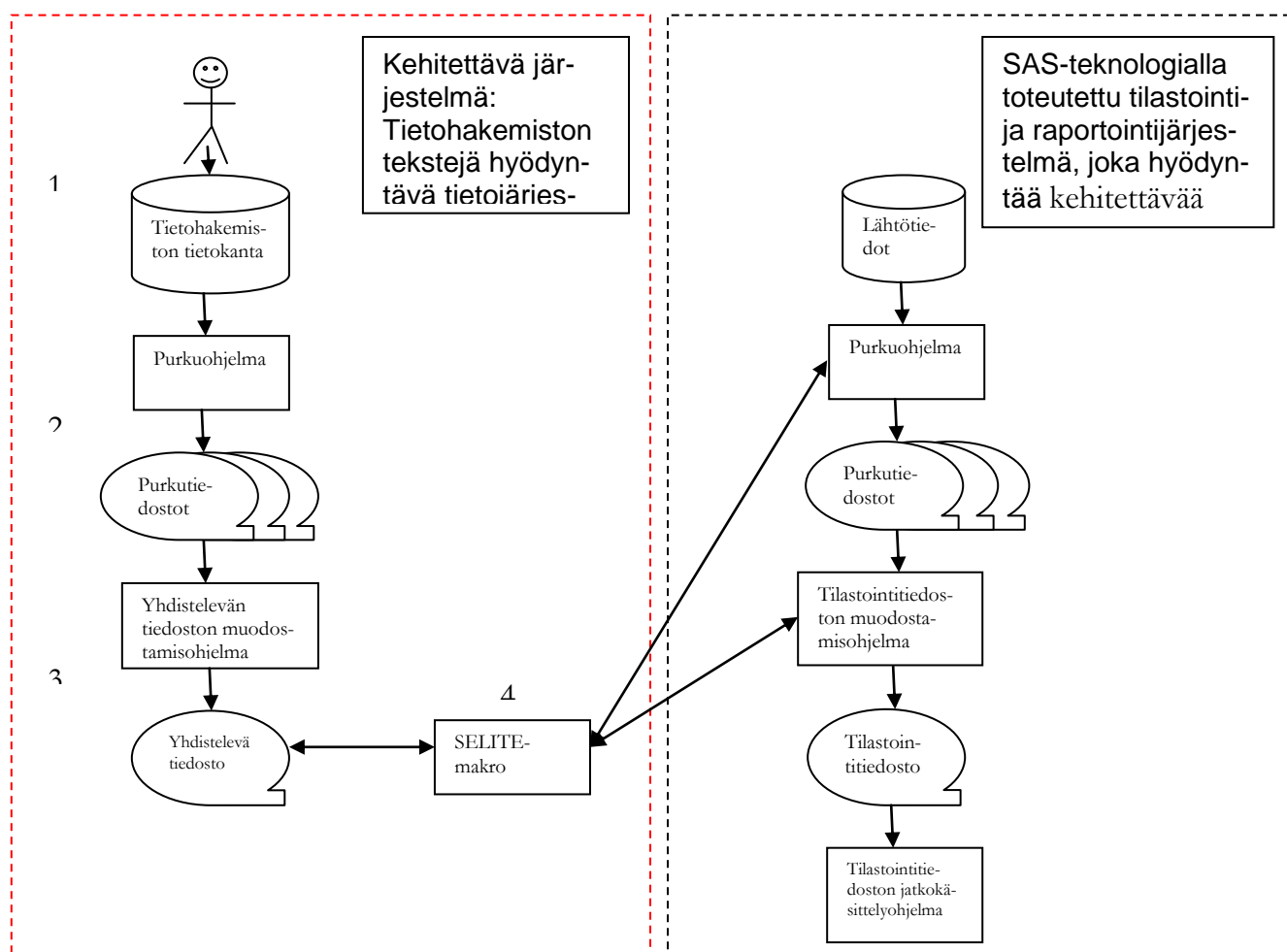
Tietojärjestelmää on voitava testata nykyhetkellä ja tulevaisuuteen, että tuottaa oikeat tekstit	2,5	Tietojärjestelmä pysyy tuottamaan ainoastaan nykyhetkeen perustuvia kuvaus-tekstejä, jolloin myöhemmin voidaan testata tulevia kuvaus-tekstejä ei voida testata lainkaan	Olemassa olevaa ratkaisua ei ole	Mahdollistetaan tietojärjestelmän testaus myös sellaisissa tapauksissa, joissa kuvaus on määritelty tulevan voimaan vasta tulevaisuudessa
Makro on tallennettava standardiohjelmiston avulla sovittuun kirjastoon	3	Makroa ei tallenneta sovittuun paikkaan, jolloin sen käyttö ja ylläpito vaikeutuvat	Yleiskäyttöiset makrot tallennetaan yleensä standardiohjelmiston avulla tiettyyn pääkoneympäristön kirjastoon	Noudatetaan makron tallennuksessa sovittuja määrittelyitä

4 Tuotteen yleiskuvaus

4.1 Tuotenäkökulma

Tietohakemiston tekstejä hyödyntävä tietojärjestelmä liittyy organisaatiossa keväällä 2012 käyttöön otettuun Tietohakemistoon. Kehitettävän tietojärjestelmän rakenne ja rajapinnat esitellään kuvassa 1.

1. Toimeksiantaja tai määrittelijä tallentaa kuvaustekstit Tietohakemiston tietokantaan, joka on DB2-pohjainen tietokanta.
2. Kehitettävään tietojärjestelmään määritellään säännöllinen Tietohakemiston tietokannan purkuajo, jonka tuloksena on purkutiedostoja.
3. Seuraavaksi purkutiedostot käsitellään tietojärjestelmään määritellyssä yhdistelytiedoston muodostamisohjelmassa, joka muodostaa selitetekstit sisältävän yhdistelevän tiedoston.
4. Tietojärjestelmään kuuluu myös SELITE-makro, joka toimii rajapintana tietojärjestelmää hyödyntäviin SAS-teknologialla toteutettaviin tilastointi- ja raportointijärjestelmiin. Sen avulla vaiheessa 3 muodostetusta yhdistelevästä tiedostosta voidaan tuottaa ohjelmallisesti selitetekstejä muiden järjestelmien käyttöön.



Kuva 1. Kehitettävän tietojärjestelmän rakenne ja rajapinnat

4.2 Ominaisuuksien yhteenveto

Tietohakemiston tekstejä hyödyntävän tietojärjestelmän ominaisuudet ja hyödyt

Asiakkaan hyöty	Tukevat ominaisuudet
SAS-tiedostojen ja -muuttujien selitteet ovat oikeita	Selitteet tuotetaan ohjelmallisesti Tietohakemiston tiedoista
SAS-tiedostojen ja -muuttujien selitteet ovat ajan tasalla	Tietohakemiston tietokannan purku suoritetaan ajastetusti vähintään kerran viikossa
Tilastointi- ja raportointijärjestelmien ylläpito helpottuu	Selitteet tuotetaan ohjelmallisesti Tietohakemistosta tiedoista
Tietojärjestelmän ylläpito ja kehittäminen on vaivatonta	Kehitystyössä noudatetaan sovittuja standardeja ja se dokumentoidaan huolellisesti

4.3 Oletukset ja riippuvuudet

Tietojärjestelmä:

- on suunniteltu nimenomaan SAS-teknologiaa käyttäville tilastointi- ja raportointijärjestelmille
- toimii Kelan pääkoneympäristössä
- vaatii toimiakseen Tietohakemiston tietokannan sisällön
- purkuajastuksen on oltava oikein määritelty
- Kelan sisäisen tiedonhuolto-prosessin on toimittava määritellyllä tavalla
- on testattu ja toteutettu SAS:n nykyisellä versiolla; toiminnallisuudet on tarkastettava versionvaihtojen yhteydessä
- rakennetaan siten, että siinä huomioidaan tulevaisuuden tarve hyödyntää tietojärjestelmää myös MikroSAS- ja SAS/EG ohjelmilla

5 Tuotteen ominaisuudet

5.1 Tietojärjestelmän käytön on oltava yksinkertaista

Tietojärjestelmän ja makron tulevien käyttäjien makrojen käyttöosaaminen vaihtelee jonkin verran. Tästä johtuen makron käyttämisen tulee olla yksinkertaista, jotta se ei jää käyttämättä hankaluutensa vuoksi. Tietojärjestelmän kehityksessä ja testauksessa kiinnitetään huomiota siihen, että sen käyttö olisi helppoa ja yksinkertaista. Käyttäjille laaditaan selkeä ohjeistus tietojärjestelmästä ja makron käytöstä.

5.2 Tietojärjestelmän on toimittava määritysten mukaisesti oikein

Tietojärjestelmän tulee tuottaa käyttäjille oikeita tietoja, jotta sitä voidaan käyttää. Virheelliset tiedot vievät järjestelmän uskottavuuden ja voivat aiheuttaa ongelmia muissa tätä järjestelmää hyödyntävissä tietojärjestelmissä.

5.3 Tietojärjestelmän kehityksessä on käytettävä standardiohjelmistoa, ja makro on tallennettava sovittuun paikkaan

Jotta tietojärjestelmän siirto eri systeemympäristöihin sekä versiointi ja lokitus toimisivat oikein, on järjestelmän kehityksessä käytettävä standardiohjelmistoa. Lisäksi ohjelma-komponenttien tallennuksessa on noudatettava sovittuja standardeja; makro on tallennettava sovittuun kirjastoon.

5.4 Tietohakemiston tietokannan purun on tapahduttava säännöllisesti ja oikein ajoitetusti, ja purku on voitava suorittaa myös tarvittaessa

Tietojärjestelmään määritellään Tietohakemiston tietokannan purku ajastetusti vähintään kerran viikossa, jotta makroa hyödyntävät sovelluskehittäjät saisivat käyttöönsä ajan tasalla olevat tiedot. Tämä palvelee myös tiedostojen loppukäyttäjiä.

5.5 Pystyy tuottamaan eri kuvausversioiden selitetekstejä

Joskus tarvitaan esimerkiksi viiden vuoden takaista selitetekstiä, tai toisaalta halutaan testata myöhemmin voimaan tulevia tekstejä. Tietojärjestelmän tulee voida tuottaa myös vanhempia tai myöhemmin voimaan tulevia selitetekstejä esimerkiksi aikaparametrin avulla.

5.6 Toiminta virhetilanteissa on suunniteltava huolellisesti

Mikäli tietojärjestelmään liittyvä SELITE-makro ei jostakin syystä onnistu tuottamaan selitetekstejä oikein kutsuvalle järjestelmälle, virhetilanne ei saa kaataa kutsuvaa ohjelmaa. Virhetilanteesta on toimitettava tieto kutsuvan järjestelmän ylläpitäjälle.

5.7 Makrokutsun koodaamisen muihin tietojärjestelmiin on oltava helppoa

Makro on toteutettava siten, ettei järjestelmän ylläpitäjien tarvitse kirjoittaa makrokutsuun parametreiksi kaikkia tiedoston muuttujien teknisiä nimiä erikseen, vaan ne tuotetaan ohjelmallisesti.

5.8 Tietojärjestelmän ylläpito on tehtävä vaivattomaksi

Kehitystyössä noudatetaan sovittuja standardeja ja se dokumentoidaan huolellisesti. Lisäksi järjestelmälle nimetään vastuuhenkilö, joka vastaa tietojärjestelmän ylläpidosta.

5.9 Tietojärjestelmää hyödynnetään muissa tietojärjestelmissä SELITE-makron avulla

Kehitettävän tietojärjestelmän ja sitä hyödyntävien tietojärjestelmien välisenä rajapintana toimii SELITE-makro. Makron avulla tietojärjestelmien kehittäjät ja ylläpitäjät voivat tuottaa SAS-tiedostojen ja -muuttujien selitetekstit Tietohakemiston tietokannasta ohjelmallisesti, jolloin tekstit ovat oikeita ja ajan tasalla.

6 Rajoitukset

Tietojärjestelmä palvelee vain SAS-teknologialla toteutettuja tilastointi- ja raportointijärjestelmiä pääkoneympäristössä.

7 Laatumittarit

Tässä vaiheessa ei ole vielä asetettu laatumittareita.

8 Tärkeys ja prioriteetti

Järjestelmän eri ominaisuuksien prioriteetit määritellään myöhemmässä vaiheessa.

9 Muut tuotteen vaatimukset

9.1 Sovellettavat standardit

Tietojärjestelmän kehitystyössä noudatetaan sovittuja standardeja:

- ohjelmistokomponenttien nimeämiset
- ohjelmistokomponenttien sijoitus ja tuotantoon siirto standardiohjelmiston avulla
- arkkitehtuuri toteutetaan tilastointi- ja raportointijärjestelmien viitearkkitehtuuria noudattaen

9.2 Järjestelmän vaatimukset

Tietojärjestelmä on suunniteltu toimimaan pääkoneympäristössä, ja sitä operoidaan SAS-teknologialla käytettävissä olevan SAS-version avulla.

9.3 Suorituskykyvaatimukset

Selitetekstejä sisältävän yhdistelytiedoston on oltava käytettävissä 24/7. Purkuajot on ajastettu suoritettaviksi kerran viikossa siten, että ne eivät osu pääkoneen kuormitushuipuihin.

9.4 Ympäristövaatimukset

Tietojärjestelmä on suunniteltu toimimaan pääkoneympäristössä, ja sitä operoidaan SAS-teknologialla käytettävissä olevan SAS-version avulla.

10 Dokumentoinnin vaatimukset

Tietojärjestelmän kehityksestä on tuotettava seuraavat dokumentit

- JAMES-menetelmän mukaiset dokumentit:
 - Sidosryhmien tarpeet
 - Visio
 - Sanasto
 - Järjestelmän yleiskuvaus
 - Tietoturvaselvitys
 - Käyttötapausmalli (kokoava)
 - Systeemitestaussuunnitelma
- Lisäksi huolehdittava, että seuraavat asiat tulevat ilmi dokumentaatiosta:
 - Järjestelmän käyttötarkoitus ja laajuus
 - Käyttäjäryhmät ja käyttötapaukset
 - Ajokaaviot (ajot ja luotavat tiedostot)
- Koodeja on kommentoitava riittävästi
- Käyttäjille on laadittava kattava ohjeistus

Tuotettavat dokumentit katselmoidaan sopivissa työryhmissä

- Esim. JAMES-dokumenttien rakenteellinen katselmointi JAMES-tuen kanssa
- Dokumenttien sisällöllinen katselmointi toimeksiantajan kanssa

Tuotetut dokumentit tallennetaan dokumenttiarkistoon projektin päätyttyä.

10.1 Asiakkaan ohjeistus

Tietojärjestelmän käyttäjille laaditaan selkeä ja kattava ohjeistus järjestelmän käytöstä. Ohjeistuksessa on huomioitava myös sellaiset henkilöt, jotka eivät ole ennen työskennelleet SAS:lla tai eivät tunne Kelan tilastointi- ja raportointijärjestelmiä. Ohjeistus laaditaan Word-dokumenttina, jossa hyödynnetään selventäviä kuvakaappauksia.

Ohjeistus laitetaan tarjolle ainakin Kelan sisäiseen verkkoon SAS-osaamisryhmän sivustolle. Lisäksi järjestelmän käyttöönoton yhteydessä järjestetään tietoiskuja, joissa esitellään ja demotaan uutta järjestelmää. Kaikkia kehitystyöhön liittyviä sidosryhmiä informoidaan järjestelmän käyttöönotosta esim. sähköpostilla.

11 A Ominaisuuksien ominaisuudet

Kappaleessa käytetyt numerolyhenteet viittaavat kappaleessa 5 esiteltyihin järjestelmän ominaisuuksiin. Kappaleen 11 sisältämiä asioita täydennetään seuraavassa iteraatiossa.

11 A.1 Tila

Seuraavassa taulukossa on esitelty ominaisuuksien tila. Kaikki ominaisuudet ovat tällä hetkellä vielä Ehdotettu-tilassa, ja ne tullaan käsittelemään ja mahdollisesti hyväksymään seuraavassa iteraatiossa.

Ehdotettu	5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9
Hyväksytty	-
Sisällytetty	-

11 A.2 Hyöty

Seuraavassa taulukossa on arvioitu kunkin ominaisuuden hyötyä sidosryhmille.

Kriittinen	5.2, 5.4, 5.5, 5.9
Tärkeä	5.1, 5.3, 5.6, 5.8
Hyödyllinen	5.7

11 A.3 Työmäärä

Ominaisuuksien toteuttamiseen kuluvan työmäärän alustavat arviot henkilötyöpäivinä ovat seuraavat:

5.1	20 htpv
5.2	10 htpv
5.3	5 htpv
5.4	2 htpv
5.5	10 htpv
5.6	10 htpv
5.7	10 htpv
5.7	20 htpv

5.7	10 htpv
Yht.	97 htpv

11 A.4 Riski

Erillisille ominaisuuksille ei vielä tässä vaiheessa kartoitettu mahdollisia riskejä, vaan mietittiin projektin yleisiä riskejä. Riskit tulevat tarkentumaan seuraavissa iteraatioissa. Projektin tässä vaiheessa tunnistettuja riskejä on esimerkiksi:

Aikatauluviivytys; projektin prioriteetti suhteessa muihin linjaorganisaation töihin on melko alhainen, joten muut tehtävät saattavat viivästyttää tämän projektin etenemistä
Tiedonhuoltoprosessin ongelmat tai puutteet; mikäli tiedonhuoltoprosessia ei noudateta sovitusti, saattavat tietohakemiston tietokantaan tallennetut kuvaustekstit olla virheellisiä tai ne saattavat puuttua kokonaan. Tämä voi tuottaa ongelmia tämän projektin toteuttamisessa.

Tietohakemiston käyttöönotto; projektin alkuvaiheessa tunnistettu riski siitä, että tietohakemistoa ei saada otettua käyttöön suunnitellusti, on nyt poistunut.

11 A.5 Vakaus

Sidosryhmien tarpeiden kartoituksen perusteella vaikuttaa siltä, että sidosryhmät ovat hyvin yksimielisiä järjestelmän toiminnallisuudesta ja ominaisuuksista. Todennäköisyys mielipiteiden muuttumiselle on tässä vaiheessa hyvin pieni.

11 A.7 Vastuuhenkilö

Seuraavassa taulukossa on hahmoteltu alustavia vastuuhenkilöitä järjestelmän ominaisuuksille:

Ehdotettu	5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9
Hyväksytty	-
Sisällytetty	-

11 A.8 Prioriteetti

Erillisille ominaisuuksille ei vielä tässä vaiheessa kartoitettu mahdollisia prioriteetteja, vaan mietittiin projektin yleistä prioriteettia suhteessa muihin projekteihin ja linjatyöhön. Projekti on prioriteetiltaan melko alhainen suhteessa muihin menossa oleviin projekteihin. Tällä voi olla vaikutusta projektin aikatauluun ja resursseihin.

28.9.2012 22:26

Liite 4. JAMES-lopputuote Sanasto

Sanasto

Tietohakemiston tekstejä hyödyntävä tietojärjestelmä

Vastuu: IT-osasto /Järjestelmäkehitysryhmä / Miia Nopanen

Versionhallinta

Versio	Päiväys	Muokkaaaja	Kommentit
0.1	30.03.2012	Miia Nopanen	Aloitettu termien kokoaminen
1.0	08.07.2012	Miia Nopanen	Hyväksytty versio seuraavien iteraatioiden käyttöön

Dokumentin tiedot

Valmistumispäivä	08.07.2012
Sijainti	C:\Users\Miia\Desktop\Opinnäytetyö\Aloitusvaihe\3_Sanasto v1.0.docx
Tulostettu	28.9.2012 22:26
Mallipohjan versio	JAMES 3.1

Sisällys

1 Johdanto	5
1.1 Dokumentin tarkoitus ja laajuus	5
1.2 Viittaukset.....	5
1.3 Yleiskuvaus	5
2 Määritelmät.....	6
2.1 Eräajo.....	6
2.2 ETL.....	6
2.3 JCL.....	6
2.4 Label (=selite).....	6
2.5 Mainframe (=pääkone, isokone, keskustietokone).....	6
2.6 Makro	6
2.7 Proc contents -proseduuri.....	6
2.8 Pääkone.....	6
2.9 SAS.....	6
2.9.1 MikroSAS	7
2.9.2 SAS/EG.....	7
2.9.3 SAS-osaamisryhmä	7
2.10 Selainpohjainen raportointijärjestelmä.....	7
2.11 Selite (= label)	7
2.12 SELITE-makro	7
2.13 Sovellusarkkitehtuuri.....	7
2.14 Systeemikoodi.....	7
2.15 TAKE-projekti	7
2.16 Tietoarkkitehtuuri	8

2.17 Tietohakemisto	8
2.18 Tilastointi- ja raportointijärjestelmät.....	8
2.19 TSO	8
2.20 Versionhallintaväline	8
2.21 WebFOCUS	8

1 Johdanto

Tämä dokumentti liittyy Tietohakemiston tekstejä hyödyntävien tietojärjestelmien kehitysprojektiin.

1.1 Dokumentin tarkoitus ja laajuus

Dokumenttiin on koottu kaikki projektin sisällön ja tavoitteiden ymmärtämisen kannalta olennaiset termit ja käsitteet selityksineen. Termistöä on koottu sitä mukaa, kun muissa projektiin liittyvissä JAMES-dokumenteissa on tullut esille uusia, määrittelyä vaativia termejä. Termien selitykset on tuotettu yhteistyössä projektin omistajan kanssa. Selitykset on laadittu erityisesti Kelan näkökulmasta.

Dokumenttia päivitetään koko projektin ajan, ja sitä voivat hyödyntää kaikki projektin toimijat sekä sidosryhmät

1.2 Viittaukset

Tämä dokumentti liittyy kaikkiin muihin em. projektin puitteissa tuotettuihin JAMES-dokumentteihin. Tämä dokumentaatio löytyy verkkolevyltä projektin kansioista.

1.3 Yleiskuvaus

Dokumentti koostuu projektissa merkityksellisistä termeistä ja niiden selityksistä. Kukin termi määritellään omassa alakappaleessaan. Lisäksi SAS-teknologiaan liittyvät termit on esitelty omassa ryhmässään.

2 Määritelmät

Seuraavissa alakappaleissa termit esitellään aakkosjärestyksessä. SAS-teknologiaan liittyvät termit on ryhmitelty omaan alaryhmäänsä.

2.1 Eräajo

Tietokoneelle annetaan joukko datan käsittelyyn liittyviä tehtäviä, jotka se suorittaa tietysti järjestyksessä peräkkäin tai rinnakkain.

2.2 ETL

ETL on lyhenne sanoista Extract-Transform-Load. Kyseessä on tiedon varastoinnin prosessi, jossa tiedot haetaan lähdejärjestelmästä, niitä muokataan ja ne tallennetaan tietovarastoon.

2.3 JCL

Job Control Language eli ajonohjauskieli. Sen avulla ohjataan pääkoneella ajettavia eräajo-ohjelmia.

2.4 Label (=selite)

SAS-tiedostoon tai SAS-muuttujaan liittyvä otsikkotieto, joka kertoo kohteen sisällön. (erityisesti SAS-ohjelmointikielessä).

2.5 Mainframe (=pääkone, isokone, keskustietokone)

Kelassa käytössä oleva IBM Z/OS -käyttöjärjestelmällä toimiva keskustietokone, joka suorittaa sille määritettyjä tehtäviä.

2.6 Makro

Jollakin ohjelmointikielellä koodattu tietokoneohjelma, joka suorittaa määritellyt tehtävät käyttäjän puolesta. SAS-teknologiassa parametroituja yleisohjelmia kutsutaan makroiksi.

2.7 Proc contents -proseduuri

SAS-ohjelmiston proseduuri, jonka avulla voidaan tutkia ja tulostaa SAS-tiedoston metadatoja (esimerkiksi tiedostossa olevien muuttujien nimet, tietomuodot ja pituudet).

2.8 Pääkone

Ks. Mainframe

2.9 SAS

SAS on lyhenne sanoista Statistical Analysis System. Kyseessä on SAS Institute Inc:n kehittämä ohjelmistoperhe ja komentokieli. SAS-ohjelmistojen päätarkoituksena on taulukkomuotoisen datan käsittely ja hyödyntäminen eri tarkoituksiin.

2.9.1 MikroSAS

SAS Institute Inc:n kehittämä työasemalla toimiva SAS-ohjelmisto.

2.9.2 SAS/EG

SAS Enterprise Guide. SAS Institute Inc:n kehittämä graafiseen käyttöliittymään perustuva SAS-ohjelmisto.

2.9.3 SAS-osaamisryhmä

Kelassa toimiva eri osastojen SAS-ammattilaisista koostuva ryhmä. Ryhmän tarkoituksena on edistää Kelan toimihenkilöiden SAS-osaamista sekä luoda SAS-teknologiaan liittyville järjestelmille standardeja.

2.10 Selainpohjainen raportointijärjestelmä

Kelan selainpohjainen raportointijärjestelmä intranetissä ja Internetissä. Sen avulla voidaan tuottaa erilaisia dynaamisia raportteja käyttäjän valintoihin perustuen.

2.11 Selite (= label)

Tiedostoon, muuttujaan tms. liittyvä otsikkotieto, joka kertoo kohteen sisällön. (erityisesti SAS-ohjelmointikielessä).

2.12 SELITE-makro

Kehitettävään tietojärjestelmään kuuluva makro (yleisohjelma), jonka avulla tilastointi- ja raportointijärjestelmien kehittäjät ja ylläpitäjät voivat tuottaa ohjelmiinsa selitetekstit ohjelmallisesti suoraan Tietohakemiston tietokannasta.

2.13 Sovellusarkkitehtuuri

Sovellusarkkitehtuuri kuvaa sovellusohjelmiston sisäisen rakenteen ja ulkoiset rajapinnat. Sovellusarkkitehtuurissa voidaan linjata myös esimerkiksi nimeämisstandardit, käytettävät teknologiat sekä tuotettavat dokumentit.

2.14 Systeemikoodi

Kelassa on käytössä tekninen tapa nimetä tietyt yhteen liittyvät asiat tietyllä koodituksella. Systeemikoodi vaikuttaa esimerkiksi tiedostojen ja ajojen nimeämiseen.

2.15 TAKE-projekti

Kelan sisäinen projekti, jonka tavoitteena oli ottaa Tietohakemisto käyttöön.

2.16 Tietoarkkitehtuuri

Tietoarkkitehtuurissa kuvataan organisaation keskeisimmät tiedot, niiden määritelmät ja riippuvuudet.

2.17 Tietohakemisto

DB2-pohjaisesta tietokannasta ja Tietohakemisto-sovelluksesta koostuva tietojärjestelmä, jonka avulla tallennetaan ja hallitaan Kelan järjestelmissä käytettävien tietoelementtien kuvauksia.

2.18 Tilastointi- ja raportointijärjestelmät

Eri teknologioilla toteutettuja järjestelmiä, joiden avulla Kelan lähdejärjestelmistä poimitaan määrittelyn mukaisia tietoja ja tuotetaan niistä summatasoisia tilastoja ja raportteja päätöksenteon tueksi.

2.19 TSO

Tekstieditoripääte, jolla voidaan olla yhteydessä Kelan pääkoneeseen.

2.20 Versionhallintaväline

Kelassa käytössä oleva ohjelmistokomponenttien versionhallinta- ja kokoonpanonhallintaväline Mainframe- eli pääkoneympäristössä.

2.21 WebFOCUS

Raportointiohjelmisto, jolla tuotetaan esimerkiksi selainpohjaisen raportointijärjestelmän raportit.

28.9.2012 22:26:00

Liite 5. JAMES-lopputuote Järjestelmän yleiskuvaus

Järjestelmän yleiskuvaus

Tietohakemiston tekstejä hyödyntävä tietojärjestelmä

Vastuu: IT-osasto /Järjestelmäkehitysryhmä / Miia Nopanen

Versionhallinta

Versio	Päiväys	Muokkaaaja	Kommentit
0.1	30.03.2012	Miia Nopanen	Aloitettu dokumentin kokoaminen
0.2	30.07.2012	Miia Nopanen	Jatkettu työstämistä
0.3	09.08.2012	Miia Nopanen	Toimeksiantajan hyväksymä versio seuraavissa iteraatioissa täydennettäväksi

Dokumentin tiedot

Valmistumispäivä
Sijainti

09.08.2012

C:\Users\Miia\Desktop\Opinnäytetyö\Aloitusvaihe\O
parin liitteiksi\4_Järjestelmän yleiskuvaus v0.3.docx

Tulostettu
Mallipohjan versio

28.9.2012 22:26
JAMES 3.1

Sisällys

1 Johdanto	4
1.1 Dokumentin tarkoitus	4
1.2 Määrittelyn laajuus	4
1.3 Määritelmät	4
2 Määrittelyperusteet	5
3 Kelan organisaatiot	5
3.1 IT-osaston tilastointi- ja raportointijärjestelmistä vastaavat asiakasvastuutiimit	5
3.2 IT-osaston tilastoinnin ja raportoinnin sovellusarkkitehtuurista vastaava tiimi	5
3.3 Tilastollisen materiaalin loppukäyttäjät, esim. IT-osaston asiakasosastot ja - ryhmät Kelan sisällä	5
4 Ulkoiset sidosryhmät	5
4.1 Tilastollisen materiaalin tilaajat	5
5 Roolityypit	6
5.1 Kehitettävän järjestelmän vastuuhenkilö ja ylläpitäjä	6
5.2 SELITE-makron (yleisohjelman) ylläpitäjä	6
5.3 Sovelluskehittäjä	6
5.4 SAS-tiedoston loppukäyttäjä	6
6 Tietojärjestelmän yleiset toiminnallisuudet	6
7 Prosessikuvaus	6
8 Järjestelmäarkkitehtuuri	7
9 Ratkaisujen perustelut	7

1 Johdanto

1.1 Dokumentin tarkoitus

Dokumentin tarkoitus on antaa yleiskäsitys Tietohakemiston tekstejä hyödyntävästä tietojärjestelmästä ja siihen liittyvistä komponenteista.

1.2 Määrittelyn laajuus

Tietojärjestelmää koskevat määrittelyt kirjataan dokumenttiin viitearkkitehtuurin mukaisesti. Määrittelyt tulevat täydentymään tulevilla iteraatioilla.

1.3 Määritelmät

Dokumentin kannalta olennaiset termit ja lyhenteet on kirjattu Sanasto-dokumenttiin, jota päivitetään koko projektin ajan. Sanasto-dokumentti on tallennettu verkkolevylle projektin kansioon.

2 Määrittelyperusteet

Tietohakemiston tekstejä hyödyntävä tietojärjestelmä on tekniseltä luonteeltaan suurta tietomäärää kerrallaan käsittelevä järjestelmä, jonka toiminta ei saa häiritä lähdejärjestelmän operatiivista toimintaa. Se toteutetaan tilastointi- ja raportointijärjestelmien viitearkkitehtuuria soveltaen. Yhtenä lopputuloksena on SELITE-makro (yleisohjelma), jota hyödyntävät järjestelmät ovat eräajojärjestelmiä.

3 Kelan organisaatiot

3.1 IT-osaston tilastointi- ja raportointijärjestelmistä vastaavat asiakasvastuutiimit

Tiimin jäsenet hyödyntävät kehitettävää tietojärjestelmää esimerkiksi eläkkeitä, perheetuksia, työttömyysturvaa, sairasturvaa sekä vammais- ja kuntoutusetuuksia koskevien tilastointi- ja raportointijärjestelmien kehittämis- ja ylläpitotyössä.

3.2 IT-osaston tilastoinnin ja raportoinnin sovellusarkkitehtuurista vastaava tiimi

Tiimin jäsenet hyödyntävät kehitettävää tietojärjestelmää esimerkiksi tietopyyntöihin ja muihin toimeksiantoihin liittyvissä tilastointi- ja raportointijärjestelmien kehittämis- ja ylläpitotyössä. Kehitettävä tietojärjestelmä on luonteeltaan työväline suhteessa muihin tilastointijärjestelmiin. Sen omistajuus on edellä mainitulla tiimillä.

3.3 Tilastollisen materiaalin loppukäyttäjät, esim. IT-osaston asiakasosastot ja -ryhmät Kelan sisällä

Tämän ryhmän edustajat toimivat usein toimeksiantajan roolissa, eli he tilaavat IT-osastolta tutkimuskäyttöön tarvittavia tilastointitiedostoja. He eivät itse todennäköisesti tule käyttämään tietojärjestelmää tai SELITE-makroa, mutta tiedostojen loppukäyttäjinä he tulevat hyötymään tietojärjestelmän toiminnallisuuden tuloksista eli oikeista ja ajan tasaisista selitteistä.

4 Ulkoiset sidosryhmät

4.1 Tilastollisen materiaalin tilaajat

Tutkijat, yritykset tai organisaatiot voivat tilata lupamenettelyn kautta Kelalta tutkimuskäyttöön tilastollista materiaalia. Mikäli materiaali toimitetaan SAS-tiedostona, voidaan sen sisältämät selitetekstit tuottaa kehitettävän tietojärjestelmän avulla.

5 Roolityypit

5.1 Kehitettävän järjestelmän vastuuhenkilö ja ylläpitäjä

IT-osaston tilastoinnin ja raportoinnin sovellusarkkitehtuurista vastaavan tiimin edustaja, joka huolehtii järjestelmän ylläpidosta.

5.2 SELITE-makron (yleisohjelman) ylläpitäjä

IT-osaston tilastoinnin ja raportoinnin sovellusarkkitehtuurista vastaavan tiimin edustaja, joka huolehtii erityisesti järjestelmään kuuluvan makron ylläpidosta.

5.3 Sovelluskehittäjä

IT-osaston tilastointi- ja raportointijärjestelmistä vastaavien asiakasvastuutiimien edustaja, joka hyödyntää tietojärjestelmän toiminnallisuutta omien järjestelmiensä ylläpitotehtävissä.

5.4 SAS-tiedoston loppukäyttäjä

IT-osaston asiakasosaston edustajat ja Kelan ulkopuoliset tilaajat, jotka käyttävät selitteitä sisältävää tiedostoa. Selitteet auttavat ymmärtämään ja tulkitsemaan tietoa oikein.

6 Tietojärjestelmän yleiset toiminnallisuudet

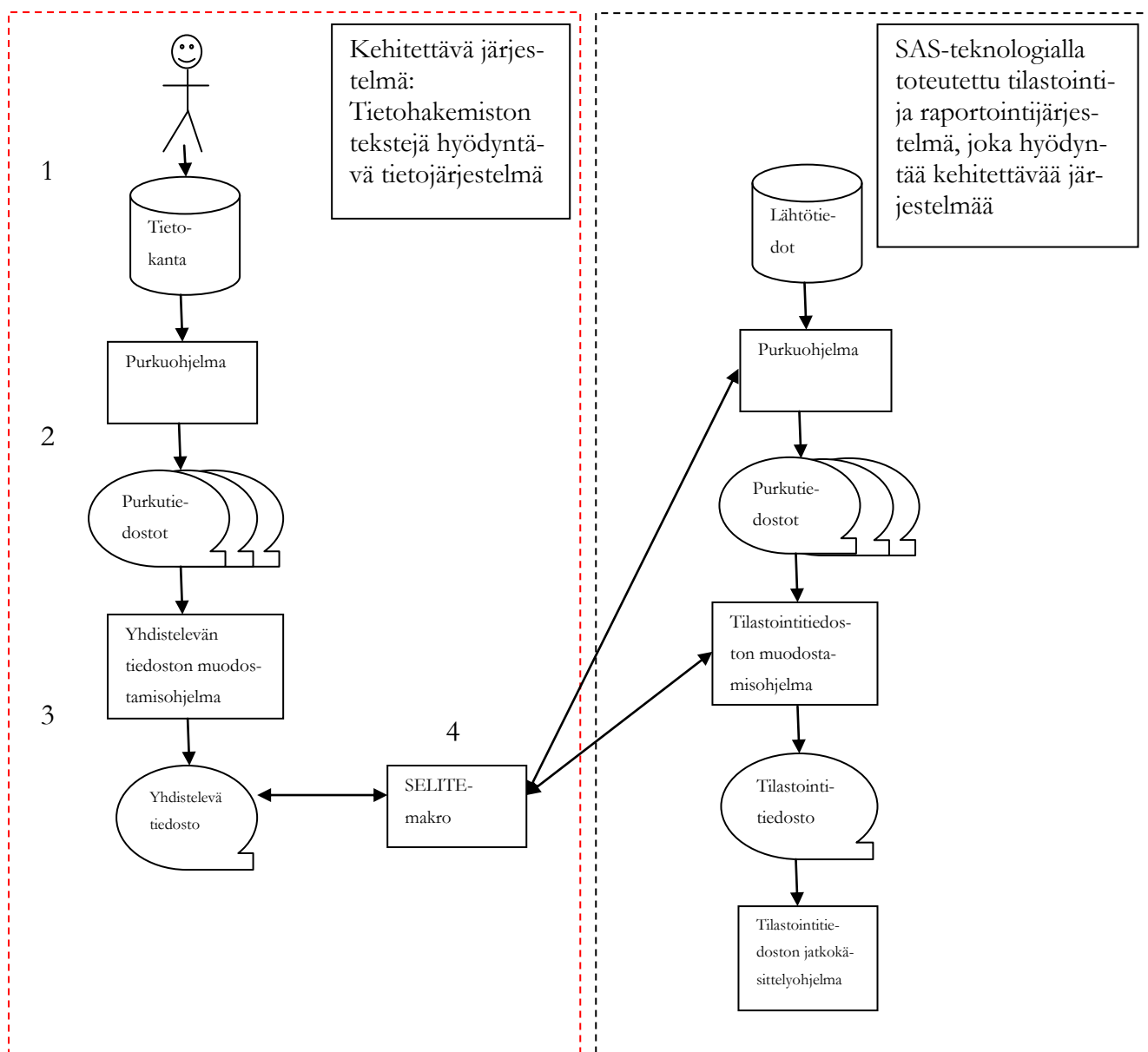
Tietojärjestelmän avulla voidaan tuottaa tilastointi- ja raportointijärjestelmien SAS-teknologialla tuotettujen tilastointitiedostojen sekä niiden sisältämien muuttujien selitetekstit eli labelit ohjelmallisesti suoraan lähdejärjestelmästä (Tietohakemiston tietokanta).

7 Prosessikuvaus

Numerointi viittaa kappaleen 8 kuvioon.

1. Toimeksiantaja tai määrittelijä tallentaa kuvaustekstit Tietohakemiston tietokantaan, joka on DB2-pohjainen tietokanta.
2. Tietojärjestelmään on määritelty säännöllinen lähdejärjestelmän tietojen purkuajovaihe, jonka tuloksena on purkutiedostoja.
3. Purkutiedostot käsitellään tietojärjestelmään määritellyssä yhdistelytiedoston muodostamisohjelmassa, joka muodostaa selitetekstit sisältävän yhdistelevän tiedoston.
4. Tilastointi- ja raportointijärjestelmien kehittäjät ja ylläpitäjät kutsuvat ohjelmistaan SELITE-makroa, joka toimii rajapintana tietojärjestelmää hyödyntäviin SAS-teknologialla toteutettaviin tilastointi- ja raportointijärjestelmiin. Sen avulla vaiheessa 3 muodostetusta yhdistelevästä tiedostosta voidaan tuottaa ohjelmallisesti selitetekstejä muiden järjestelmien käyttöön.

8 Järjestelmäarkkitehtuuri



9 Ratkaisujen perustelut

Ratkaisuja täsmennetään ja perustellaan seuraavassa iteraatiossa.

28.9.2012 22:26:00

Liite 6. JAMES-lopputuote Tietoturvaselvitys

Tietoturvaselvitys

Tietohakemiston tekstejä hyödyntävä tietojärjestelmä

Vastuu: IT-osasto / Järjestelmäkehitysryhmä / Miia Nopanen

Versionhallinta

Versio	Päiväys	Muokkaaaja	Kommentit
0.1	30.03.2012	Miia Nopanen	Aloitettu dokumentin kokoaminen
0.2	03.05.2012	Miia Nopanen	Jatkettu työstämistä
0.3	09.05.2012	Miia Nopanen	Tehty muokkauksia katselmoinnin pohjalta
0.4	24.05.2012	Miia Nopanen	Tehty viimeistelyt yhteiskatselmoinnin pohjalta
1.0	30.07.2012	Miia Nopanen	Toimeksiantajan hyväksymä versio seuraavien iteraatioiden käyttöön.

Dokumentin tiedot

Valmistumispäivä	30.07.2012
Sijainti	C:\Users\Miia\Desktop\Opinnäytetyö\Aloitusvaihe\Oparin liitteiksi\5_Tietoturvaselvitys v1.0.docx
Tulostettu	28.9.2012 22:26
Mallipohjan versio	JAMES 3.1

Sisällys

1 Johdanto	4
1.1 Dokumentin tarkoitus	4
2 Tietoturvan kartoittaminen	5
2.1 Järjestelmän tiedot	5
2.1.1 Järjestelmän tuottama data	5
2.1.2 Järjestelmän komponentit	5
2.2 Tietoturvasaso	6
2.3 Käyttöoikeuksien hallinta	6
2.4 Rekisteriseloste (julkisuuslaki 621/1999)	6
2.5 Tunnistus- ja todennusmenettelyt	6
2.6 Sidosryhmien tietoturvalle asettamat vaatimukset	7
2.7 Seuranta ja valvonta	7
2.8 Varmuuskopiointi ja palauttaminen	8
3 Tarvittavat jatkotoimenpiteet	8
4 Tietoturvaselvityksen tilanne	8

1 Johdanto

1.1 Dokumentin tarkoitus

Tämä dokumentti liittyy Tietohakemiston tekstejä hyödyntävien tietojärjestelmien kehitysprojektiin. Dokumentissa kuvataan tietojärjestelmän kehitystyössä sekä valmiissa tietojärjestelmässä huomioitavat tehtävät ja toimenpiteet tietoturvan varmistamiseksi.

Dokumentin rakenne noudattelee JAMES-menetelmän mukaista mallipohjaa, ja tarkentavat kysymykset on jätetty näkyviin tekstin lomaan.

Tämän dokumentin ymmärtämisen kannalta olennaiset termit ja lyhenteet on kirjattu Sanasto-dokumenttiin.

Dokumentissa kuvatut tietoturvaan liittyvät vaatimukset kirjataan tiedoksi myös Visio-dokumenttiin.

2 Tietoturvan kartoittaminen

2.1 Järjestelmän tiedot

2.1.1 Järjestelmän tuottama data

- **Miten suojataan tahalliselta ja tahattomalta muuttumiselta?**
Tietojärjestelmän tuottamat yhdistelytiedostot suojataan tahattomalta muokkaukselta ja poistamiselta siten, että ne tallennetaan tuotantoympäristöön lukuoikeuksin, jotka on määritelty erikseen sovittavalle käyttäjäryhmälle.

Kehitettävälle tietojärjestelmälle sovitaan ennen käyttöönottoa vastuuhenkilöt, joiden tehtävänä on päivittää ja ylläpitää järjestelmää. Vastuuhenkilöt seuraavat järjestelmän toimintaa ja sen tuottamien tietojen oikeellisuutta mm. käyttöön otetun sovelluksen häiriötilanteissa. Tietojärjestelmän käyttäjä ohjeistetaan siitä, että he eivät saa tehdä muutoksia järjestelmään. Tietojärjestelmään kuuluvat ohjelmistokomponentit viedään tuotantoympäristöön standardiohjelmiston kautta, jolloin versiointi tapahtuu automaattisesti ja komponenttien muokkaamisesta jää tieto järjestelmään. Mikäli järjestelmää muutetaan tahattomasti jonkun käyttäjän toimesta, on tilanteesta tiedotettava vastuuhenkilöä, joka voi tarvittaessa palauttaa edellisen version.

- **Miten järjestelmän tiedot suojataan tiedonsiirron aikana esim. lähiverkossa?**
Järjestelmä toimii Mainframe-ympäristössä, joten tämä kohta ei koske kyseistä tietojärjestelmää.
- **Miten järjestelmän tiedot suojataan säilytettäessä esim. tietokannassa?**
Tiedot tallennetaan tietojärjestelmän tuottamiin SAS-muotoisiin yhdistelytiedostoihin tuotanto-ympäristössä. Tiedostoihin määritellään vain lukuoikeudet sisältävät käyttövaltuudet.
- **Miten oikeudetonta tietojen katselua pyritään ehkäisemään?**
Tietoja ei tarvitse suojata katselulta, sillä ne eivät sisällä henkilötietoja tai muita arkaluontoisia tietoja. Tiedot ovat julkisia Kelan sisällä. Kuvaustekstejä sisältäviä tiedostoja ei luovuteta sellaisenaan Kelan ulkopuolelle, vaan tekstejä hyödynnetään muissa Kelan ulkopuolelle luovutettavissa tiedostoissa ainoastaan järjestelmään kuuluvan makron avulla. Järjestelmän vastuuhenkilö vastaa käyttäjien ohjeistamisesta.

2.1.2 Järjestelmän komponentit

- **Kuka saa katsoa / käyttää järjestelmää ja missä yhteydessä?**
Tietojärjestelmän ja siihen sisältyvien komponenttien (ml. tietojärjestelmän tuottamat yhdistelytiedostot) katselu sekä järjestelmän käyttö on sallittu kaikille sovelluskehitystyötä tekeville Kelan toimihenkilöille.

Tietojärjestelmään sisältyvä makro sijoitetaan makroille tarkoitettuun kirjastoon, josta sitä voidaan kutsua muista ohjelmista.

Mikäli lähdejärjestelmän tietokannan purku on tarpeen suorittaa ajastettujen purkuajojen välillä, käyttäjät voivat tehdä sen kerta-ajopyyntönä. Tietokannan luonteen vuoksi purkuajon voi tilata suoritettavaksi myös päivääikaan.

- **Ohjelmistojen päivitysten ja tietoturvapäivitysten prosessikuvaus?**
Tietojärjestelmään liittyvien komponenttien päivityksestä huolehtivat nimetyt vastuuhenkilöt. Järjestelmän käyttäjiä ohjeistetaan, etteivät he tee itsenäisesti tietojärjestelmän komponentteihin päivityksiä. Mahdollisista muutostarpeista informoidaan järjestelmän vastuuhenkilöä, joka suorittaa toivotut muutokset ohjelmistokomponentteihin.

2.2 Tietoturvasato

Kehitettävän järjestelmän lainmukainen tietoturvasato?
Tietoturva-asetuksen 681/2010 ja sitä täsmentävän VAHTI-ohjeen 2/2010 mukaan järjestelmän tietoturvasatoksi riittää perustaso, sillä kyseessä on koodisto- / apurekisterityyppistä tekstietoa käsittelevä tietojärjestelmä, joka ei sisällä henkilötietoa.

2.3 Käyttöoikeuksien hallinta

- **Onko roolipohjainen?**
Kyllä, käyttöoikeudet anotaan käyttövaltuusjärjestelmän kautta.
- **Roolien kuvaukset ja oikeudet?**
Tilastointi- ja raportointijärjestelmien kanssa työskentelevät IT-osaston henkilöt voivat käyttää järjestelmän tietoja lukuoikeuksin.

2.4 Rekisteriseloste (julkisuuslaki 621/1999)

- **Miten on huomioitu?**
Rekisteriselostetta ei laadita, koska kyseessä on koodisto- / apurekisterityyppistä tekstietoa käsittelevä tietojärjestelmä, joka ei sisällä henkilötietoa.

2.5 Tunnistus- ja todennusmenettelyt

- **Onko käytössä salasana ja käyttäjätunnistus menetelmä, vai jokin muu, mikä?**
Tietojärjestelmän käyttäjät kirjautuvat järjestelmään omilla käyttäjätunnuksillaan.
- **Salasanan pituus ja kompleksisuusvaatimukset?**
Kelan määritysten mukaiset.

- **Miten ylläpitäjän salasana suojataan järjestelmässä, entä tiedonsiirron aikana esim. lähiverkossa?**
Ylläpitäjällä ei ole erillistä käyttöliittymää eikä salasanaa, vaan jokainen kirjautuu omilla tunnuksillaan.
- **Miten käyttäjän salasana suojataan järjestelmässä, entä tiedonsiirron aikana esim. lähiverkossa?**
Mainframe-ympäristö hoitaa tarvittavat suojaukset.

2.6 Sidosryhmien tietoturvalle asettamat vaatimukset

- **Onko sidosryhmillä erityis- tai lisävaatimuksia tietoturvalle (salattavat tiedot, julkiset tiedot)?**
Ei lisävaatimuksia.

2.7 Seuranta ja valvonta

- **Mitä lokitetaan, lokien tietosisältö?**
Mainframe-ympäristön versionhallintaohjelmisto pitää lokia käyttäjien toimenpiteistä tietojärjestelmäkomponenttien osalta. Tietosisältö on versionhallintaohjelmiston määritysten mukainen. Lisäksi ohjelmisto kirjoittaa tuotantoympäristön datan käsittelystä lokia sisäiseen arkistojärjestelmään.
- **Miten säilytetään?**
Versionhallintaohjelmiston määritysten mukaisesti.
- **Säilytysaika?**
Versionhallintaohjelmiston määritysten mukainen.
- **Kuka saa katsoa ja missä yhteydessä?**
Versionhallintaohjelmiston määritysten mukaisesti, ei erityisrajoituksia.
- **Onko automaattihälytyksiä käytössä, millaisia?**
Ei ole.
- **Miten suojataan tahalliselta ja tahattomalta muuttumiselta?**
Versionhallintaohjelmiston määritysten mukaisesti.
- **Miten oikeudetonta lokitietojen katselua pyritään ehkäisemään?**
Versionhallintaohjelmiston määritysten mukainen.
- **Onko yksittäisellä käyttäjällä esim. ylläpitäjällä mahdollisuus muuttaa loki-tietoja siten, että muutoksista ei jää jälkeä?**
Ei ole.
- **Miten lokien kellonajat tahdistetaan?**
Versionhallintaohjelmiston määritysten mukaisesti.

2.8 Varmuuskopiointi ja palauttaminen

- **Miten huolehditaan ja kuka vastaa?**

Järjestelmän komponentit

Versionhallintaohjelmiston huolehtii tietojärjestelmän komponenttien versioinnista. Vastuuhenkilöt hoitavat palauttamisen, mikäli tarpeen.

Järjestelmän tuottama data

Tiedostot määritellään versionhallintaohjelmistoon siten, ettei niistä tuoteta sukupolvitiedostoja. Tiedostojärjestelmä huolehtii tiedostojen säilytyksestä.

3 Tarvittavat jatkotoimenpiteet

Todetut vaatimukset kirjataan tiedoksi myös Visio-dokumenttiin.

Tietoturvaselvitysdokumenttia päivitetään projektin edetessä tarpeen mukaan.

4 Tietoturvaselvityksen tilanne

Tietoturvaselvitys on valmis aloitusvaiheen osalta, ja sitä päivitetään tarpeen vaatiessa tulevilla iteraatioilla.

28.9.2012 22:26:00

Liite 7. JAMES-lopputuote Systemitestisuunnitelma

Systemitestisuunnitelma

Tietohakemiston tekstejä hyödyntävä tietojärjestelmä

Vastuu: IT-osasto / Järjestelmäkehitysryhmä / Miia Nopanen

Versionhallinta

Versio	Päiväys	Muokkaaaja	Kommentit
0.1	08.07.2012	Miia Nopanen	Aloitettu dokumentin kokoaminen
0.2	09.08.2012	Miia Nopanen	Toimeksiantajan hyväksymä versio seuraavissa iteraatioissa täydennettäväksi

Dokumentin tiedot

Valmistumispäivä	09.08.2012
Sijainti	C:\Users\Miia\Desktop\Opinnäytetyö\Aloitusvaihe\Oparin liitteiksi\6_Systeemitestisuunnitelma v0.2.docx
Tulostettu	28.9.2012 22:26
Mallipohjan versio	JAMES 3.1

Sisällys

1 Johdanto	4
1.1 Dokumentin tarkoitus.....	4
1.2 Yhteydet muihin dokumentteihin	4
1.3 Suunnitelman päivitysvastuu	4
2 Testauksen tavoitteet.....	5
3 Testauksen tehtävät	5
3.1 Testauskohteet	5
3.2 Suunnitellut testit	5
4 Testausmenettelyt.....	5
4.1 Käyttötapaukset	5
4.2 Testisuunnitelmat.....	6
4.3 Testin apuvälineet	6
5 Keskeiset projektin vaiheet ja testauksen alustava aikataulu	6
6 Raportointi.....	6
7 Testaajat ja resurssit.....	6
8 Avoimet asiat	7

1 Johdanto

Tämä dokumentti liittyy Tietohakemiston tekstejä hyödyntävien tietojärjestelmien kehitysprojektiin. Tietohakemiston tekstejä hyödyntävän tietojärjestelmän tarkoitus on mahdollistaa SAS-tiedostojen ja -muuttujien selitetekstien tuottaminen ohjelmallisesti Tietohakemiston tietokannasta.

1.1 Dokumentin tarkoitus

Dokumentissa kuvataan kyseisen tietojärjestelmän systeemitestauksen tavoitteet, tehtävät, testausmenettelyt sekä alustava aikataulu, raportointivastuu, testauksen perustana olevat dokumentit ja testauksen resurssit. Testaussuunnitelman tarkoituksena on kuvata yleisellä tasolla systeemitestauksen suunnitteluun, toteutukseen, toimintatapoihin ja hallintaan vaikuttavat keskeiset perusperiaatteet. Suunnitelma pyrkii kattamaan koko projektin elinkaaren testauksen näkökulmasta.

1.2 Yhteydet muihin dokumentteihin

Systeemitestaussuunnitelman lähtöaineistona ovat seuraavat dokumentit:

- Visio
- Kokoava käyttötapausmalli; käyttötapauskuvaukset (ei vielä valmis)

Lisäksi siinä hyödynnetään seuraavia dokumentteja:

- Sanasto

1.3 Suunnitelman päivitysvastuu

Suunnitelmaa päivitetään projektipäällikön tai tehtäväkokonaisuudesta vastaavan henkilön toimesta koko projektin ajan. Päivitystarve tarkistetaan vähintään jokaisen iteraation suunnittelemisen yhteydessä.

2 Testauksen tavoitteet

Testauksen tavoitteena on toteuttaa systeemitestaus suunnitellussa aikataulussa ja varmistaa, että Tietohakemiston tekstejä hyödyntävä tietojärjestelmä toimii virheettömästi ja määriteltujen vaatimusten mukaisesti.

3 Testauksen tehtävät

Testauksen tehtävänä on suunnitella ja toteuttaa tietojärjestelmälle ja sen komponenteille testit, joilla todetaan sovelluksen vastaavan asetettuja vaatimuksia ja toimivan määrittelyn mukaisesti. Samalla testataan, että sovellus integroituu ympäristöön sovitulla tavalla.

3.1 Testauskohteet

Testauksen kohteita ovat mm.:

- makron toiminnallisuus; onko helppokäyttöinen ja vaatimusmäärittelyn mukainen
- makron palauttaman tiedon oikeellisuus; tuottaako oikean selitetekstin (testattava myös aikaparametroituna)
- vikasetokyky; aiheuttaako kutsuvan ohjelman kaatumisen virhetilanteessa, jos ei pysty tuottamaan tarvittavia selitetekstejä

3.2 Suunnitellut testit

Testauksessa suoritetaan ohjelma- ja aineistotestausta, ja se sisältää tietojärjestelmän toiminnallisuuden, käytettävyyden ja luotettavuuden testausta.

4 Testausmenettelyt

4.1 Käyttötapaukset

Testaus pohjautuu käyttötapauksiin ja niihin liittyviin vaatimuksiin. Käyttötapaus on kirjallinen kuvaus yhdestä järjestelmän toiminnallisuudesta käyttäjän näkökulmasta. Käyttötapaus voidaan ilmaista myös käyttäjäroolin ja järjestelmän välisenä vuoropuheluna.

Käyttötapauksella on esiehto, jonka on oltava voimassa, jotta käyttötapaus voidaan suorittaa. Jos tämä ehto ei täyty, ei käyttötapauksia suoriteta. Edelleen käyttötapauksella on oltava lopputila eli mihin tilaan järjestelmä päättyy käyttötapauksen valmistuttua.

Käyttötapauksilla testataan sekä normaali tapahtumien kulku eli tapahtumaketjua, joka yleisimmin järjestelmässä toteutuu sekä vaihtoehtoiset tapahtumien kulut, joka edustavat yleensä poikkeusta normaalista tapahtumien kulusta. Myös virhekäsittelyt kuuluvat vaihtoehtoisin tapahtumankulkuihin.

Jokaisesta käyttötapauksesta johdetuista vaatimuksista luodaan testitapaukset, joilla testataan käyttötapauksen oikea toiminta ja todennetaan järjestelmän vaatimukset. Viimeistään rakennusvaiheessa testataan testitapauksen kattaman määritellyn toiminnan lisäksi sitä, mitä ohjelman ei kuulu tehdä. Tämä tehdään rakentamalla testitapaukset täydentävi

en vaatimusten testaamiseen. Myös näistä ei-toiminnallisista vaatimuksista pitäisi löytyä jäljitettävyyttä ominaisuusvaatimuksiin toiminnallisten vaatimusten kaltaisesti.

Jokaisesta vaatimuksesta tehdään ainakin yksi testitapaus. Jokaisesta testitapauksesta kirjoitetaan lisäksi testikuvaus (testiskripti), jossa kuvataan testin suoritus yksityiskohtaisesti, vaihe vaiheelta. Pyrkimyksenä on kirjoittaa kuvaukset niin yksityiskohtaisesti, että testitapauksen pystyy suorittamaan myös muu kuin testitapauksen suunnittelija.

4.2 Testisuunnitelmat

Tietojärjestelmän kehittäjä laatii testisuunnitelmat ja -tapaukset, kun käyttötapaukset on määriteltä ja kuvattu.

4.3 Testin apuvälineet

Testausta suoritetaan omilla työasemilla käyttäen hyväksi standardiohjelmistoja.

Järjestelmästä havaitut virheet kirjataan sovitulla verkkolevyllä sijaitsevaan Excel-tilukkuun.

5 Keskeiset projektin vaiheet ja testauksen alustava aikataulu

Projektin vaiheet ja testauksen aikataulu täsmentyvät myöhemmin.

6 Raportointi

Kehitystyön loppuvaiheessa toteutetussa testauksessa havaitut virheet raportoidaan järjestelmän kehittäjälle (virheet tai ongelmat kirjataan esimerkiksi sovitussa sijainnissa olevaan Excel-tilukkuun).

7 Testaajat ja resurssit

Tietojärjestelmän kehittäjä suorittaa ohjelma- ja aineistotestausta jatkuvasti kehitystyön aikana. Hän varmistaa, että järjestelmä toimii teknisesti oikein.

Toimeksiantaja suorittaa systeemitestauksen. Hän vastaa siitä, että järjestelmä voidaan ottaa käyttöön. Toimeksiantajan testaustyö sijoittuu projektin loppuvaiheeseen lähelle käyttöönottoa.

Lisäksi hyvissä ajoin ennen käyttöönottoa pyydetään yhtä tai kahta makron tulevaa käyttäjää testaamaan, että makro ja sen taustalla oleva tietojärjestelmä täyttää sille asetetut vaatimukset. Havaitut vaatimusmäärittelyn vastaiset virhetoiminnallisuudet korjataan ennen käyttöönottoa. Muut mahdolliset kehitysehdotukset dokumentoidaan ja niiden toteuttamista harkitaan tulevissa ohjelmistoversioissa.

8 Avoimet asiat

Ennen kuin testaussuunnitelmaa voidaan työstää eteenpäin, on projektissa laadittava käyttötapausmalli-dokumentti.